

Wavelets

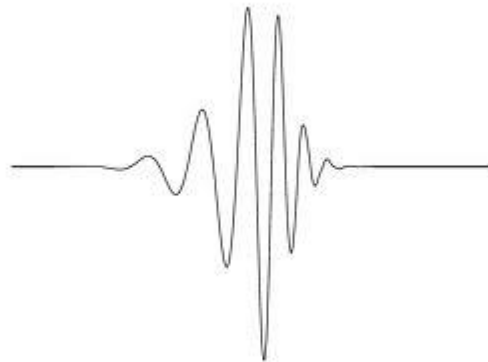
Proyecto Sistemas Informáticos

Facultad de Informática

Universidad Complutense de Madrid

Profesor Director: Luis Vázquez

Elena Rupérez Cerezo



RESUMEN

Este trabajo constituye principalmente una panorámica de los fundamentos y aplicación de la teoría de wavelets. Comenzamos con una base matemática para luego dar una introducción general de los conceptos sobre las series de Fourier, Transformada de Fourier, Transformada Discreta de Fourier DFT y Transformada Rápida de Fourier FFT. Posteriormente se introduce wavelets como una herramienta alternativa al análisis de Fourier para el procesamiento de señales.

La idea es desarrollar de una manera clara la teoría matemática de wavelets y su aplicación en las siguientes áreas que a continuación describo: Imágenes comprimidas con wavelets, texturas de imágenes digitales, mejoras como por ejemplo resaltar el contorno de una imagen. Compresores basados en la transformada Wavelet, comparación con aplicaciones tan actuales como Zip. Aplicaciones de la transformada Wavelet en el campo del álgebra lineal.

SUMMARY

This work is primarily an overview of the fundamentals and application of the theory of wavelets. We begin with a mathematical basis then give a general introduction to the concepts of Fourier series, Fourier Transform, Discrete Fourier Transform DFT and FFT Fast Fourier Transform. Wavelet is an alternative tool to Fourier analysis for signal processing.

The idea is to develop a clear way the mathematical theory of wavelets and their application in the following areas that I describe below: Images compressed with wavelet texture digital image enhancements such as highlight the outline of an image. Compressors based on wavelet transform, as compared to current applications as Zip. Wavelet transform applications in the area of linear algebra.

Índice General

1. Introducción
2. Base Matemática – Conceptos
3. Teoría de Fourier
 - 3.1 Series de Fourier
 - 3.2 Transformada de Fourier
 - 3.3 Transformada Discreta de Fourier (DFT)
 - 3.4 Transformada Rápida de Fourier (FFT)
4. Análisis Tiempo – Frecuencia
 - 4.1 La Transformada corta de Fourier (STFT)
 - 4.2 Transformada Continua Wavelet (CWT)
5. Sistemas Wavelet
 - 5.1 Función Escala
 - 5.2 Función Wavelet
6. Wavelet Haar
 - 6.1 Principios de Multi-resolución
 - 6.2 Función Haar Escala
 - 6.3 Función Haar Wavelet
 - 6.4 Transformada Discreta Wavelet (DWT)
 - 6.5 Transformada rápida wavelet. Banco de filtros, Ejemplo
7. Aplicaciones
 - 7.1 Aplicaciones de la DWT en la solución de sistemas de ecuaciones lineales.
 - 7.2 Compresión
 - 7.2.1 Compresión de Imagen con MatLab 7.0
 - 7.2.2 Implementación compresión de imágenes en .NET C#.
 - 7.2.3 Aplicaciones de la transformada de wavelet en mejora de imágenes.
 - 7.2.4 Compresor Wavelet vs. ZIP
 - 7.2.5 JPEG vs JPE2000

8. Conclusiones

9. Bibliografía

10 Referencias Internet

11 Apéndice

1. Introducción

Un sistema Wavelet puede describirse de forma “amigable”, como un conjunto de ladrillos (que para cada sistema pueden tener diferente forma), los cuales sirven para reconstruir o representar una función o señal. Este conjunto es una expansión bi-dimensional $L^2(\mathbb{R})$, que suele servir de base para cualquier clase de señal de una o más dimensiones.

Realizaremos un recorrido por las actuales componentes matemáticas más conocidas y daremos paso y cabida a las nuevas componentes que hay en el mercado explicando su teoría y sus posibles aplicaciones.

2. Base Matemática

En este apartado vamos a ir añadiendo todos aquellos conceptos matemáticos que nos serán de utilidad para entender los razonamientos que vamos a exponer a continuación:

Ortogonalidad entre el seno y el coseno

Se dice que las funciones del conjunto $\{f_k(t)\}$ son ortogonales en el intervalo $a < t < b$ si dos funciones cualesquiera $f_m(t)$, $f_n(t)$ de dicho conjunto cumplen:

$$\int_a^b f_m(t)f_n(t)dt = \begin{cases} 0 & \text{para } m \neq n \\ r_n & \text{para } m = n \end{cases}$$

Ortonormal – Normalidad

Un conjunto de vectores es ortonormal si es a la vez un conjunto ortogonal y la norma de cada uno de sus vectores es igual a 1.

Definición 1.2.1 Un sistema ortogonal de H es un subconjunto de elementos no nulos del espacio ortogonales dos a dos, i.e. si e_i, e_j son dos elementos del sistema y $i \neq j$, entonces $\langle e_i, e_j \rangle = 0$.

Si además son unitarios ($\|e_i\| = 1, \forall e_i$) decimos que el sistema es ortonormal.

Diremos que el sistema ortonormal $E \subset H, E = \{e_i\}_{i \in I}$ es **completo** si $E^\perp = \{0\}$, i.e. si $\langle x, e_i \rangle = 0, \forall i \in I$ implica que $x = 0$. Un sistema ortonormal completo se llama también **base hilbertiana** o **base ortonormal** de H .

Espacios vectoriales

Un espacio vectorial sobre un cuerpo K (como el cuerpo de los números reales o los números complejos) es un conjunto V no vacío, dotado de dos operaciones para las cuales será cerrado:

$$\begin{aligned} \text{Suma } + : V \times V &\longrightarrow V \\ (\mathbf{u}, \mathbf{v}) &\mapsto \mathbf{u} + \mathbf{v} \end{aligned}$$

Operación interna tal que:

- 1) tenga la propiedad conmutativa, es decir
 $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}, \forall \mathbf{u}, \mathbf{v} \in V$
- 2) tenga la propiedad asociativa, es decir
 $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}, \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in V$
- 3) tenga elemento neutro $\mathbf{0}$, es decir
 $\exists \mathbf{0} \in V : \mathbf{u} + \mathbf{0} = \mathbf{u}, \forall \mathbf{u} \in V$
- 4) tenga elemento opuesto, es decir
 $\forall \mathbf{u} \in V, \exists -\mathbf{u} \in V : \mathbf{u} + (-\mathbf{u}) = \mathbf{0}$

y la operación producto por un escalar:

$$\begin{aligned} \text{Producto } \cdot : K \times V &\longrightarrow V \\ (a, \mathbf{u}) &\mapsto a \cdot \mathbf{u} \end{aligned}$$

Operación externa tal que:

- 5) tenga la propiedad asociativa:
 $a \cdot (b \cdot \mathbf{u}) = (a \cdot b) \cdot \mathbf{u}, \forall a, b \in K, \forall \mathbf{u} \in V$
- 6) tenga elemento neutro 1:
 $\exists 1 \in K : 1 \cdot \mathbf{u} = \mathbf{u}, \forall \mathbf{u} \in V$
- 7) tenga la propiedad distributiva del producto respecto la suma de vectores:
 $a \cdot (\mathbf{u} + \mathbf{v}) = a \cdot \mathbf{u} + a \cdot \mathbf{v}, \forall a \in K, \forall \mathbf{u}, \mathbf{v} \in V$
- 8) tenga la propiedad distributiva del producto respecto la suma de escalares:
 $(a + b) \cdot \mathbf{u} = a \cdot \mathbf{u} + b \cdot \mathbf{u}, \forall a, b \in K, \forall \mathbf{u} \in V$

Señales Estacionarias

La primera división natural de todas las señales es en las categorías estacionarias y no estacionarias. Las señales estacionarias son constantes en sus parámetros estadísticos sobre tiempo. Si uno observa una señal estacionaria, durante unos momentos y después espera una hora y vuelve a observar, esencialmente se vería igual, eso es, su nivel general sería casi lo mismo y su distribución de amplitud y su **desviación estándar** serían casi lo mismo. La maquinaria rotativa generalmente produce señales de vibración estacionarias.

Las señales estacionarias se dividen en señales deterministas y aleatorias. Las señales aleatorias son impredecibles en cuanto a su contenido de **frecuencia** y a su nivel de amplitud, pero todavía tienen características estadísticas relativamente uniformes sobre tiempo.

Ejemplos de señales aleatorias son lluvia cayendo en un techo, ruido de un motor a reacción, turbulencia en los patrones de flujo de una bomba y **cavitación**.

Teorema de Parseval

En matemáticas, la Relación de Parseval demuestra que la Transformada de Fourier es unitaria; es decir, que la suma (o la integral) del cuadrado de una función es igual a la suma (o a la integral) del cuadrado de su transformada. Esta relación procede de un teorema de 1799 sobre series, cuyo creador fue Marc Antoine Parseval. Esta relación se aplicó más tarde a las Series de Fourier.

En física e ingeniería, la Relación de Parseval se suele escribir como:

$$\int_{-\infty}^{\infty} |f(t)|^2 dt = \int_{-\infty}^{\infty} |\mathcal{F}[f(t)](\alpha)|^2 d\alpha$$

donde $\mathcal{F}[f(t)](\alpha)$ representa la transformada continua de Fourier de $x(t)$ y f representa la frecuencia (en hercios) de x .

Discretizar

Dado un dominio $\Omega \subset \mathbb{R}^d$ con una frontera continua en el sentido de Lipschitz una partición en n "elementos finitos", es una colección de n subdominios $\{\Omega^{(e)}\}_{e=1}^n$ que satisface:

1. $\Omega = \bigcup_{e=1}^n \Omega^{(e)}$
2. Cada $\Omega^{(e)}$ es un conjunto compacto con una frontera Lipschitz-continua.
3. $\text{int}(\Omega^{(i)}) \cap \text{int}(\Omega^{(j)}) = \emptyset, \quad i \neq j$

Multi-resolución

Puede observarse que los objetos de una imagen resultan identificables empleando diferentes escalas de ampliación. En general, una aproximación multiresolucional aplicada a la representación o el análisis de una imagen intenta aplicar este concepto.

Un ejemplo muy ilustrativo podemos encontrarlo en la cartografía. La escala de un mapa está dada por la relación entre la dimensión real del territorio y su representación sobre el papel. Para escalas elevadas, solamente resultan visibles las características más macroscópicas, mientras que los detalles solamente pueden emerger para escalas más bajas. En las wavelets este aspecto se realiza dilatando y contrayendo la wavelet madre para crear el conjunto de las funciones base. La wavelet madre para valores elevados nos da las características extensas, frente a valores pequeños de la función se analizan los detalles de la señal.

Compresión m:n

Se define la ESCALA como la relación entre la dimensión dibujada respecto de su dimensión real, esto es:

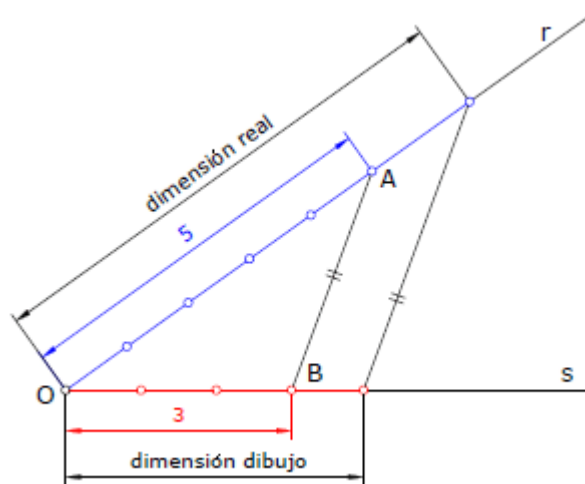
$$E = \text{dibujo} / \text{realidad}$$

Véase, por ejemplo, el caso para E 3:5

1º) Con origen en un punto O arbitrario se trazan dos rectas r y s formando un ángulo cualquiera.

2º) Sobre la recta r se sitúa el denominador de la escala (5 en este caso) y sobre la recta s el numerador (3 en este caso). Los extremos de dichos segmentos son A y B.

3º) Cualquier dimensión real situada sobre r será convertida en la del dibujo mediante una simple paralela a AB.



Identidad de Euler

La identidad es un caso especial de la Fórmula de Euler, la cual especifica que

$$e^{ix} = \cos x + i \sin x$$

para cualquier número real x. (Nótese que los argumentos para las funciones trigonométricas sen y cos se toman en radianes.) En particular si

$$x = \pi$$

entonces

$$e^{i\pi} = \cos \pi + i \sin \pi$$

y ya que

$$\cos \pi = -1$$

y que

$$\sin \pi = 0$$

se sigue que

$$e^{i\pi} = -1$$

Lo cual implica la identidad

$$e^{i\pi} + 1 = 0$$

3. Teoría de Fourier

Un proceso físico puede ser descrito en el dominio del tiempo mediante valores representados por una cantidad f como función del tiempo o en el dominio de la frecuencia mediante una serie de amplitudes representadas como función de la frecuencia. La transformada de Fourier es una herramienta con la capacidad de representar este proceso, o cualquier otro, tanto en el dominio del tiempo como en el dominio de la frecuencia. Esto hace que la Transformada de Fourier sea ampliamente utilizada en aplicaciones en el campo de la ciencia e ingeniería (Tratamiento de imágenes, reducción de ruido, análisis de frecuencias, análisis de materiales y estadísticas, síntesis...etc.).

La ventaja de los wavelets frente a la transformada de Fourier es su soporte compacto. La posibilidad de realizar un análisis en un intervalo finito manteniendo su diferenciabilidad y su regularidad. Tomemos como ejemplo las ondas que producen los sonidos de una orquesta. Estas señales no son infinitas en el dominio del tiempo sino que tienen un inicio y un fin. Si tomamos un intervalo finito de una señal continua vemos que en ese intervalo no podemos aplicar la transformada de Fourier debido a que la función no es derivable en todos los puntos, pero si podremos aplicar wavelets debido a su soporte compacto.

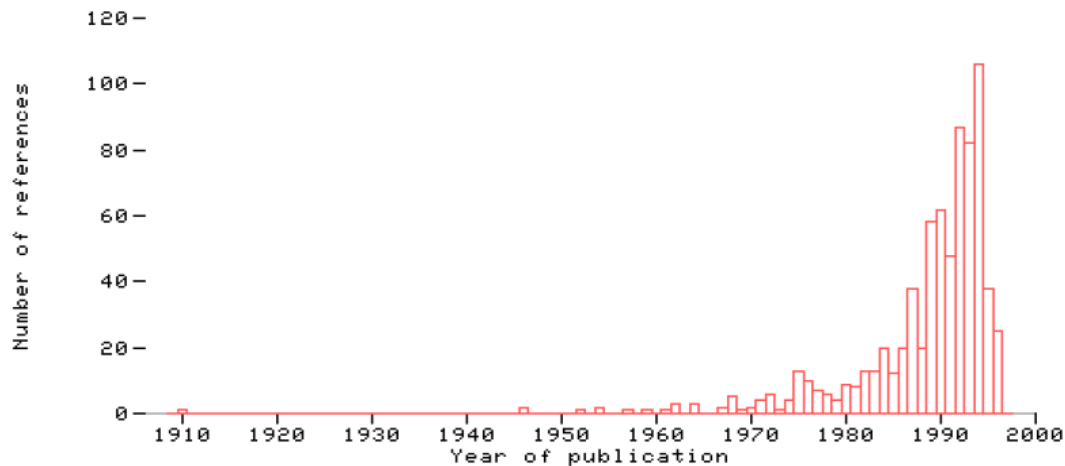


Figura 4. Número de publicaciones por año del tema Wavelet.

3.1 Series de Fourier

Estas series tienen su origen en el siglo IXI y deben el nombre a su creador Joseph Baptiste Fourier. La idea básica de las series de Fourier es que una función periódica puede ser representada como una suma ponderada de senos y cosenos.

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \{a_n \cos(nt) + b_n \sin(nt)\}$$

Esta serie toma el nombre de serie de Fourier cuando es posible obtener todos los coeficientes a_n y b_n mediante una integración de la función $f(t)$ de la siguiente manera

$$a_0 = \frac{1}{\pi} \int_0^{2\pi} f(t) dt \quad (n = 0, 1, 2, \dots)$$

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(t) \cos(nt) dt \quad (n = 0, 1, 2, \dots)$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(t) \sin(nt) dt \quad (n = 0, 1, 2, \dots)$$

Además es necesario cumplir con la condición de que la norma de la función analizada sea integrable y esa integral sea finita, es decir, la función debe ser periódica:

$$\int_0^{2\pi} |f(t)| < \infty$$

3.2 Transformada de Fourier



La transformada de Fourier, en esencia, descompone o expande una señal o función, en senos y cosenos de diferentes frecuencias cuya suma corresponde a la señal original, es decir, es capaz de distinguir las diferentes componentes de frecuencia de la señal, y sus respectivas amplitudes. La transformada de Fourier de una función del tiempo $f(t)$ se define como:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{i\omega t} dt$$

y la transformada inversa de Fourier, como:

$$f(t) = \int_{-\infty}^{\infty} F(\omega)e^{-i\omega t} d\omega$$

De acuerdo con lo dicho anteriormente la transformada de Fourier puede obtener un representación en el dominio de la frecuencia de una señal que se encuentra originalmente en el dominio del tiempo. La relación existente entre la representación de la señal original a través de funciones senoidales y cosenoidales y la exponencial proviene de la definición de la identidad de Euler.

$$e^{i\omega t} = \cos(\omega t) + i\sin(\omega t)$$

$$e^{-i\omega t} = \cos(\omega t) - i\sin(\omega t)$$

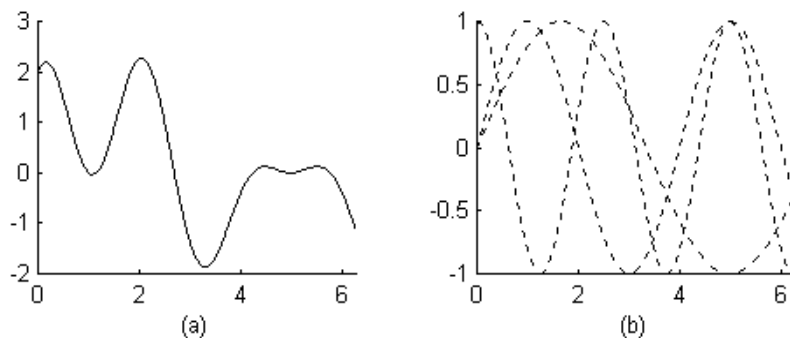
$$f(t) = F_0 + F_1 e^{-i\omega t} + F_2 e^{-i2\omega t} + F_3 e^{-i3\omega t} + \dots + F_{-1} e^{i\omega t} + F_{-2} e^{i2\omega t} + F_{-3} e^{i3\omega t} + \dots$$

$$f(t) = \sum_{-\infty}^{\infty} F_n e^{-in\omega t}$$

$$F_n = \frac{1}{T} \int_{t_0}^{t_0+T} f(t) e^{-in\omega t} dt$$

$$f(t) = \frac{a_0}{2} + \sum \{a_n \cos(n\omega t) + b_n \sin(n\omega t)\}$$

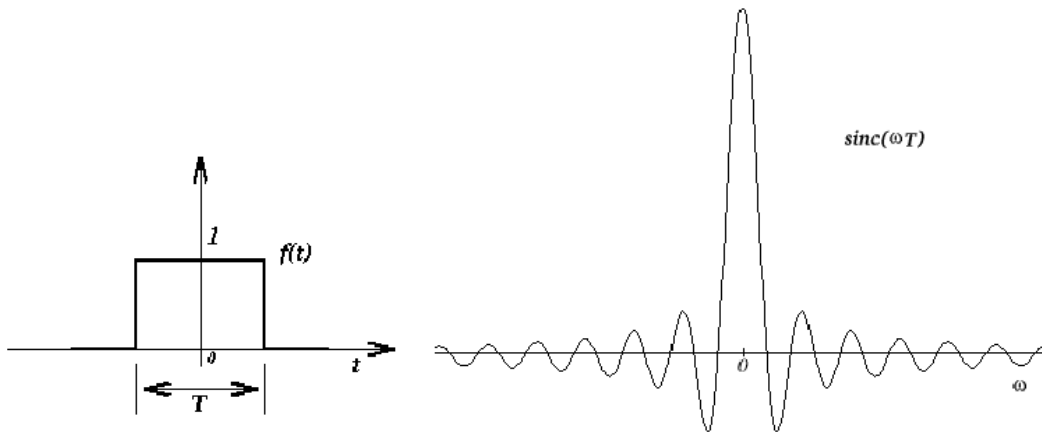
En la figura (a) podemos ver la señal original y en la figura (b) podemos ver la descomposición en series de Fourier, la amplitud de cada onda es lo que representa la transformada de Fourier.



Aunque la función exponencial resulta más fácil de manipular vamos a trabajar con funciones senos y cosenos ya que resulta más fácil comprender el paso de la señal del dominio del tiempo al dominio de la frecuencia y viceversa.

A continuación muestro un ejemplo de cómo quedaría representada la misma función en el ámbito del tiempo frente al ámbito de la frecuencia.

$$f(t) = \begin{cases} 1 & -\frac{T}{2} < t < \frac{T}{2} \\ 0 & \text{caso contrario} \end{cases}$$



Como ya es sabido, la transformada de Fourier constituye una herramienta mediante la cual podemos obtener información sobre cómo está distribuida la energía de una señal a través de sus distintas componentes de frecuencia, es decir, podemos conocer todas las componentes de frecuencia existentes en la señal y sus respectivos aportes energéticos. Se puede resumir diciendo que la transformada de Fourier tiene una perfecta resolución en frecuencia y una muy pobre resolución en tiempo.

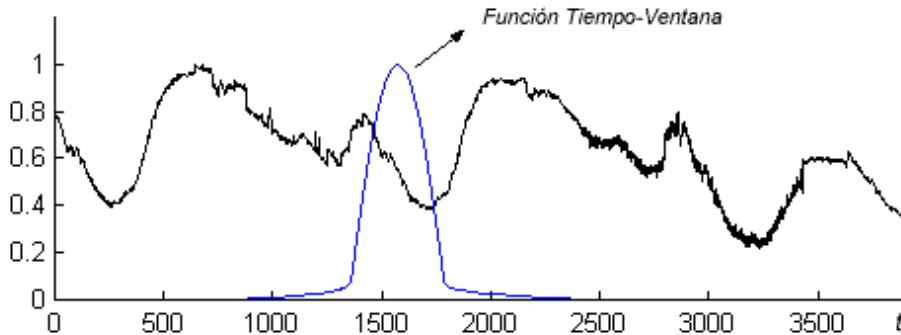
Por resolver el problema de resolución en tiempo de la transformada de Fourier, Denis Gabor (1946) adaptó la transformada utilizando un procedimiento llamado **ventanamiento**. Este procedimiento consiste en dividir una señal en pequeños segmentos a través del tiempo de tal manera que podamos asumir que para cada segmento la señal es estacionaria y así calcular la Transformada de Fourier clásica para cada porción de la señal.

La forma de dividir la señal se realiza mediante lo que llamaremos una función tiempo- ventana $h(t)$ cuyo ancho o soporte corresponde a la longitud de cada segmentación de la señal. Con la función ventana encuadramos la señal alrededor de un instante de tiempo τ y calculamos su transformada de Fourier, luego trasladamos la función ventana hasta que no se sobrepone con la anterior cubriendo una nueva porción de la señal a la que volvemos a calcular su transformada de Fourier. Este proceso es repetido hasta que se ha cubierto la totalidad de la señal.

Ahora bien, el soporte de la ventana constituye un parámetro de gran importancia ya que a través de este podemos establecer el grado de resolución tanto de tiempo como de frecuencia que deseemos. Si nuestra ventana es muy angosta analizaremos una porción muy pequeña de la señal lo que nos permite tener una buena resolución en tiempo pero una mala resolución en frecuencia ya que conoceremos sólo una mínima fracción del espectro total existente en la señal. Por otro lado, si nuestra ventana es muy ancha tendremos una buena resolución en frecuencia pero una mala resolución en tiempo.

¿Es posible que la función ventana tenga un soporte dinámico y no estático?, es decir, una función ventana que tenga la capacidad de cambiar su soporte en forma automática dependiendo del contenido espectral del segmento de la señal analizado, ya que una situación ideal de análisis sería tener una buena resolución en tiempo para frecuencias altas y una buena resolución en frecuencia frente a contenido de frecuencias bajas. Para la resolución de este problema existe una herramienta

matemática denominada la transformada continua wavelet que fue desarrollada como una alternativa de análisis frente a la Transformada Corta de Fourier STFT.



3.3 Transformada Discreta de Fourier (DFT)

Debido a que los computadores trabajan sólo con datos discretos, el cálculo numérico de la transformada de Fourier requiere valores discretos. Esto significa que mediante el uso de un computador es posible calcular la transformada sólo para valores discretos.

$F(t)$ es una señal periódica de periodo T y solo conocemos sus valores en N puntos. Tenemos:

$$w_s = 2\pi f_s$$

$$f_k = f\left(k\frac{T}{N}\right) \quad \wedge \quad T_s = \frac{T}{N}$$

$$F_n = F(nw_s) \quad \wedge \quad w_s = \frac{2\pi}{T}$$

$$F_n = \sum_{k=0}^{N-1} f_k e^{\frac{i2\pi kn}{N}} \quad n, k = 0, 1, 2, \dots, N-1$$

$$f_k = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-\frac{i2\pi kn}{N}} \quad n, k = 0, 1, 2, \dots, N-1$$

A manera de ejemplo vamos a suponer que tenemos una señal muestreada en cuatro puntos:

$$f = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

Entonces, si desarrollamos la DFT de f_k obtenemos 4 ecuaciones distintas:

$$F_0 = [f_0 e^{\frac{i2\pi 0 \cdot 0}{4}} + f_1 e^{\frac{i2\pi 0 \cdot 1}{4}} + f_2 e^{\frac{i2\pi 0 \cdot 2}{4}} + f_3 e^{\frac{i2\pi 0 \cdot 3}{4}}]$$

$$F_1 = [f_0 e^{\frac{i2\pi 0 \cdot 1}{4}} + f_1 e^{\frac{i2\pi 1 \cdot 1}{4}} + f_2 e^{\frac{i2\pi 2 \cdot 1}{4}} + f_3 e^{\frac{i2\pi 3 \cdot 1}{4}}]$$

$$F_2 = [f_0 e^{\frac{i2\pi 0 \cdot 2}{4}} + f_1 e^{\frac{i2\pi 1 \cdot 2}{4}} + f_2 e^{\frac{i2\pi 2 \cdot 2}{4}} + f_3 e^{\frac{i2\pi 3 \cdot 2}{4}}]$$

$$F_3 = [f_0 e^{\frac{i2\pi 0 \cdot 3}{4}} + f_1 e^{\frac{i2\pi 1 \cdot 3}{4}} + f_2 e^{\frac{i2\pi 2 \cdot 3}{4}} + f_3 e^{\frac{i2\pi 3 \cdot 3}{4}}]$$

$$F_0 = [f_0 + f_1 + f_2 + f_3]$$

$$F_1 = [f_0 + f_1 \{ \cos(\frac{\pi}{2}) + i \sin(\frac{\pi}{2}) \} + f_2 \{ \cos(\pi) + i \sin(\pi) \} + f_3 \{ \cos(\frac{3\pi}{2}) + i \sin(\frac{3\pi}{2}) \}]$$

$$F_2 = [f_0 + f_1 \{ \cos(\pi) + i \sin(\pi) \} + f_2 \{ \cos(2\pi) + i \sin(2\pi) \} + f_3 \{ \cos(3\pi) + i \sin(3\pi) \}]$$

$$F_3 = [f_0 + f_1 \{ \cos(\frac{3\pi}{2}) + i \sin(\frac{3\pi}{2}) \} + f_2 \{ \cos(3\pi) + i \sin(3\pi) \} + f_3 \{ \cos(\frac{9\pi}{2}) + i \sin(\frac{9\pi}{2}) \}]$$

$$F_0 = [f_0 + f_1 + f_2 + f_3]$$

$$F_1 = [f_0 + f_1(i) + f_2(-1) + f_3(-i)]$$

$$F_2 = [f_0 + f_1(-1) + f_2 + f_3(-1)]$$

$$F_3 = [f_0 + f_1(-i) + f_2(-1) + f_3(i)]$$

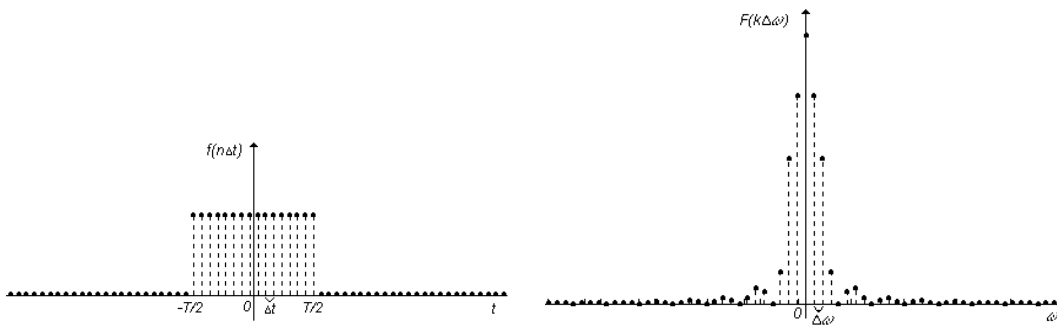
$$i = \sqrt{-1} \quad i^2 = -1 \quad i^3 = -i \quad i^4 = 1$$

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & (i) & (i)^2 & (i)^3 \\ 1 & (i)^2 & (i)^4 & (i)^6 \\ 1 & (i)^3 & (i)^6 & (i)^9 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

$$\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & (-i) & (-i)^2 & (-i)^3 \\ 1 & (-i)^2 & (-i)^4 & (-i)^6 \\ 1 & (-i)^3 & (-i)^6 & (-i)^9 \end{bmatrix} \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

El punto importante aquí es que ambas matrices tienen la misma forma con la única diferencia de un cambio de signo. Por lo tanto si nosotros podemos realizar la transformada rápidamente también podemos obtener la inversa en forma rápida entre los coeficientes y los valores de la función.

Ahora vamos a representar la misma función utilizada en el apartado anterior representando la función de forma discreta.



3.4 Transformada rápida de Fourier (FFT)

Con el fin de implementar en forma práctica la Transformada Discreta de Fourier mediante el uso de computadores, a mediados de la década del sesenta J.W Cooley y J.W Tukey desarrollaron un algoritmo denominado la Transformada rápida de Fourier (FFT). La FFT elimina información redundante que existe en la DFT.

En la práctica, la Transformada de Fourier se calcula con un algoritmo llamado “Fast Fourier Transform”. Por ejemplo en Matlab la función para calcular la transformada se llama “fft”.

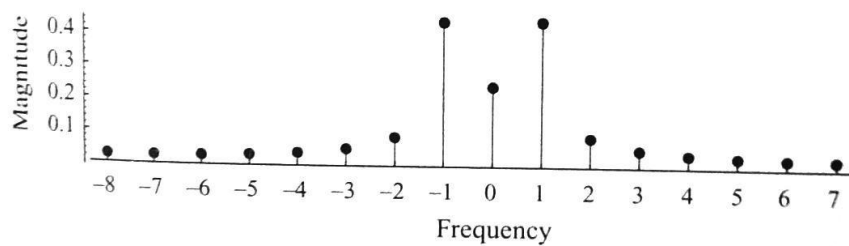
4 Análisis Tiempo – Frecuencia

4.1 La Transformada corta de Fourier (STFT)

$$\text{STFT} \{x[n]\} \equiv X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n}$$

Discontinuidades en el Tiempo

Las discontinuidades periódicas en la señal producen un espectro con muchos armónicos en altas frecuencias. La DFT “escucha” clicks en estas discontinuidades.



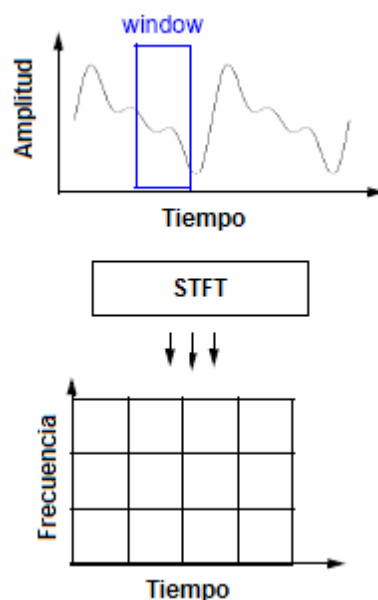
Uno de los problemas del STFT es que tiene una resolución fija. El ancho de la función de ventana está relacionado con él como la señal es representada, esto determina si hay buena resolución en frecuencia (las componentes de frecuencia que están cerca pueden ser separadas) o buena resolución en tiempo (el tiempo en cuyas frecuencias cambian). Una ventana amplia da una mejor resolución en frecuencia pero también una pobre resolución en el tiempo. Una ventana angosta da una buena resolución en el tiempo pero una pobre resolución en frecuencia. Estas son llamadas transformadas de banda angosta y de banda amplia, respectivamente.

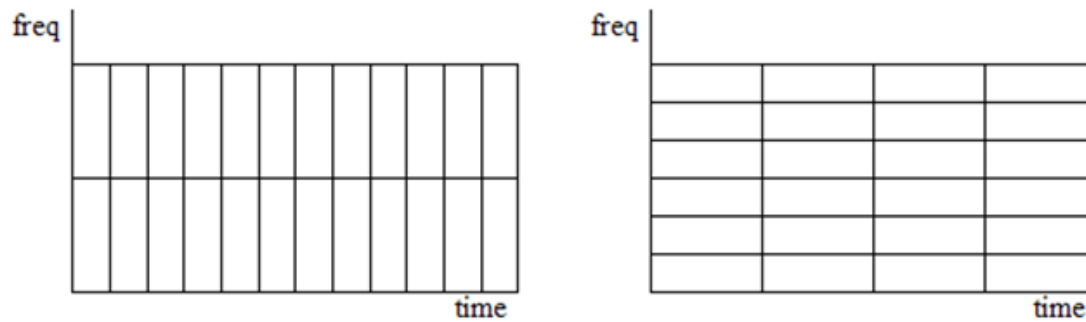
Obtener mayor resolución en frecuencia requiere un largo de ventana M mayor.

Obtener mayor resolución temporal requiere un largo de ventana M menor.

Los 2 problemas son:

- Resolución en Frecuencia: Sólo vemos las frecuencias que son múltiplos de nuestra frecuencia de análisis.
- Escape (leakage): Discontinuidades en los bordes de la ventana de análisis producen esparcimiento de ruido en el resto del espectro.

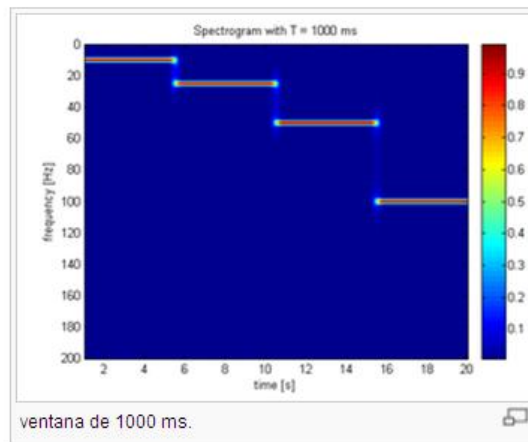
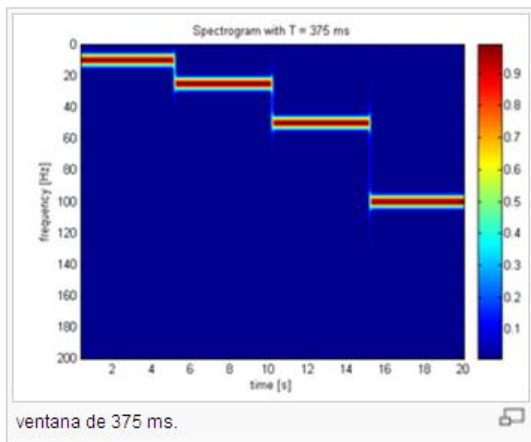
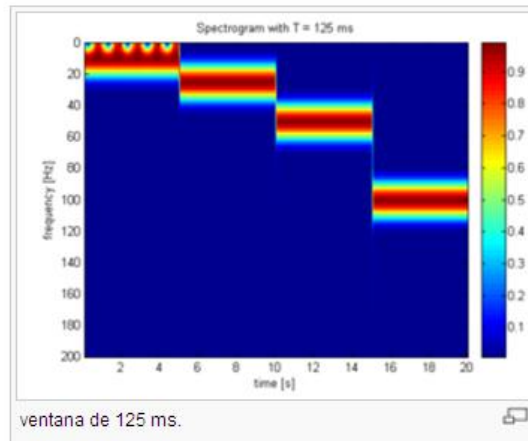
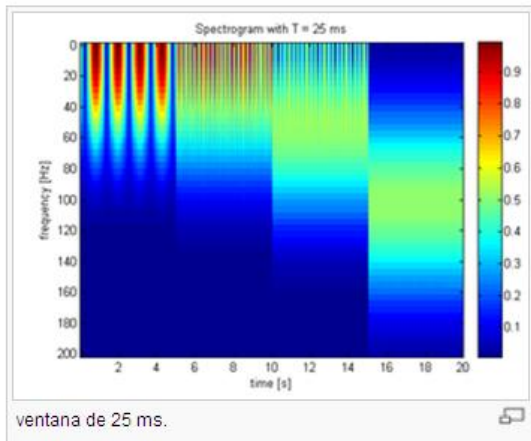




Usando la siguiente muestra de señal $x(t)$ que está compuesta por un conjunto de 4 formas de onda sinusoidales unidas en secuencia. Cada forma de onda está únicamente compuesta de una de cuatro frecuencias (10, 25, 50, 100 Hz). La definición de $x(t)$ es.

$$x(t) = \begin{cases} \cos(2\pi 10t); & 0 \leq t < 5s \\ \cos(2\pi 25t); & 5 \leq t < 10s \\ \cos(2\pi 50t); & 10 \leq t < 15s \\ \cos(2\pi 100t); & 15 \leq t < 20s \end{cases}$$

Entonces es muestreado a 400 Hz. El siguiente espectrograma fue producido:



La ventana de 25 ms nos permite identificar un tiempo preciso en el cual la señal cambia pero los cambios precisos en la frecuencia son difíciles de identificar. En el otro extremo de la escala, la ventana de 1000 ms permite que las frecuencias sean vistas de forma precisa pero el tiempo entre los cambios de frecuencia es borroso.

4.2 Transformada Continua Wavelet (CWT)

La transformada wavelet constituye una técnica relativamente nueva que ha sido propuesta por los investigadores como una poderosa herramienta en el análisis sobre el comportamiento local de una señal.

La transformada continua wavelet intenta expresar una señal $x(t)$ continua en el tiempo, mediante una expansión de términos o coeficientes proporcionales al producto interno entre la señal y diferentes versiones escaladas y trasladadas de

una función prototipo más conocida como $\psi(t)$ wavelet madre.

Dado que la nueva función wavelet madre es infinita podemos expresar la transformada continua de la siguiente manera:

$$CWT(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt$$

Ahora expresamos las funciones en términos de la transformada de Fourier.

$$CWT(a, b) = \frac{1}{2\pi\sqrt{a}} \int_{-\infty}^{\infty} X(\omega) \Psi(a\omega) e^{-i\omega b} d\omega$$

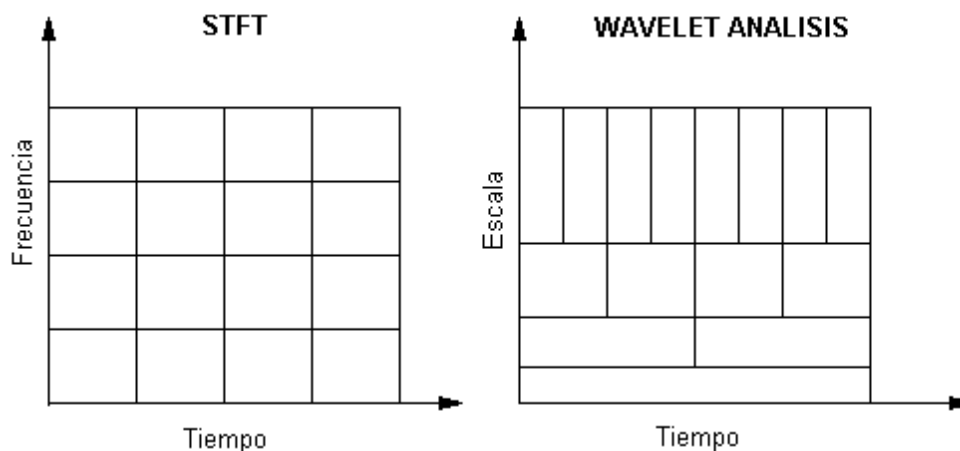
La variable a (escala) controla el ancho de banda mientras que la variable b (traslación) nos da ubicación en el dominio del tiempo. Para conseguir una reconstrucción de la señal a través de la función wavelet madre esta debe de cumplir la condición de admisibilidad.

$$\Psi(0) = 0$$

El cumplimiento de esta condición nos lleva obligatoriamente a pensar que sea una onda.

Mediante la variable la escala nosotros podemos comprimir o dilatar la señal el cual marcará el grado de resolución con el cual estamos analizando la señal. Para escalas pequeñas la CWT nos entrega una buena resolución en el dominio del tiempo mientras que para escalas grandes la CWT nos entrega una buena resolución en el dominio de la frecuencia. En lo anteriormente dicho se encuentra la diferencia principal entre la Transformada continua de Wavelet y la Transformada corta de Fourier, ya que la primera ocupa ventanas de corta duración para altas frecuencias y ventanas de larga duración para bajas frecuencias mientras que la STFT ocupa una sola ventana con la misma duración tanto para altas frecuencias como para bajas frecuencias.

Diferencia tiempo - frecuencia v/s tiempo -escala entre la STFT y la CWT.



**Compresión de imágenes
JPG vs JPG-2000**

Image with 4096 FFT coefficients



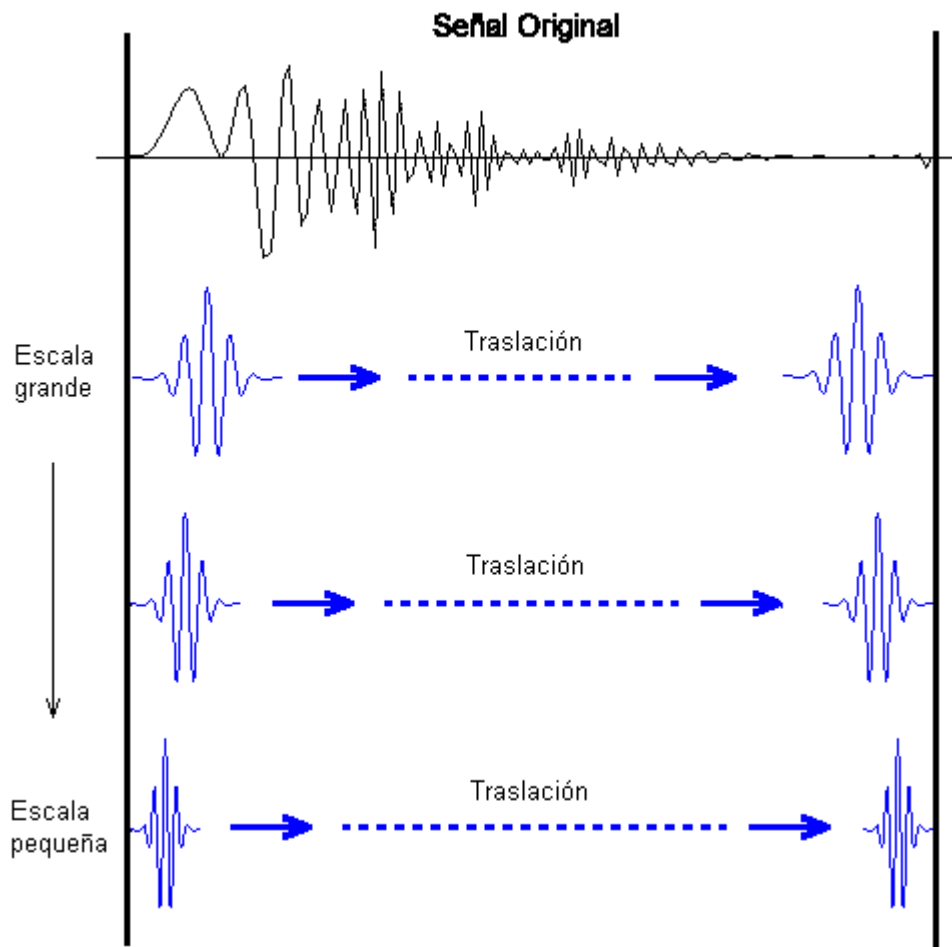
Image with 4096 DWT coefficients



Podemos llegar a realizar un cambio de variable a al dominio de la frecuencia:

$$a \rightarrow \omega = \frac{c}{a}$$

Con este cambio de variable podemos observar que la CWT localiza tanto la señal $x(t)$ en el dominio del tiempo como su espectro ω en el dominio de la frecuencia en forma simultánea.



Se observa el proceso de escalamiento y traslación, para una escala grande la wavelet ocupa un mayor segmento de la señal y por lo tanto tiene mejor resolución en frecuencia mientras que para una escala más pequeña el intervalo de tiempo bajo el que se analiza la señal es menor, lo que implica mayor resolución en tiempo.

La continuidad de la CWT reside en que tanto la variable de escala como la variable de traslación varían en forma continua. Sin embargo, en términos de cálculo computacional es imprescindible discretizar la transformada, y la suposición más lógica es que tanto los valores de escala como traslación sean discretos. Adelantándonos un poco a lo que es la Transformada Discreta Wavelet, la forma más común de discretizar los valores de a y b es utilizar una red diádica [BUR98] [CHU97].

$$a = 2^{-j} \quad b = k2^{-j}$$

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad a, b \in R, a \neq 0$$

$$\psi_{j,k}(t) = 2^{\frac{j}{2}} \psi(2^j t - k) \quad j, k \in \mathbb{Z}$$

La Transformada Continua posibilita el mapeo grafico de la información de la señal, donde pueden visualizarse las estructuras, patrones y fenómenos en el dominio tiempo - frecuencia. Esta aplicación es invariante por traslaciones o modulaciones. Sin embargo, ella requiere un considerable esfuerzo computacional y no nos provee, eficientemente, información numérica útil para realizar un profundo análisis. La Transformada Discreta, en cambio nos permite desplegar toda la información en átomos, esto es, los coeficientes en wavelets, sin redundancia.

El tratamiento con wavelets discretas permite su aplicación directa a procesos computacionales. Las wavelets continuas presentan por una parte la dificultad de su manejo al tener que evaluar un gran número de integrales y tener en consecuencia una redundancia de información, pero por otra parte permiten la flexibilidad de poder adaptarse a situaciones en las que las discretas no dan un resultado satisfactorio. Concretamente en el tratamiento de imágenes digitales se precisa realzar detalles y detectar texturas analizando la imagen desde distintos ángulos, lo cual es posible hacer con las wavelets continuas al disponer de un mayor número de parámetros que posteriormente se pueden discretizar para su tratamiento computacional.

5 Sistemas Wavelets

Los sistemas wavelet de primera generación son todos aquellos que sean generados sólo por traslaciones enteras y escalamientos de una única función wavelet), llamada wavelet madre o wavelet generadora:

$$\psi_{j,k}(t) = 2^{\frac{j}{2}} \psi(2^j t - k) \quad j, k \in \mathbb{Z}$$

Wavelet madre

$$\psi(t).$$

Función Escala

$$\phi(t).$$

Con estas dos funciones podremos aproximar cualquier función o señal $f(t)$, mediante una de las funciones o mediante ambas, de la forma:

$$f(t) = \sum_k \sum_j c_{j,k} \phi(t) + \sum_k \sum_j d_{j,k} \psi(t)$$

5.1 Función Escala

Sea $\phi(t) \in L^2(R)$, una función escala, que trasladada y escalada genera una familia de funciones definida como:

$$\phi_{j,k}(t) = 2^{\frac{j}{2}} \phi(2^j t - k) \quad j, k \in Z$$

Entonces la función $f(t)$ se puede escribir como:

$$f(t) = \sum_{k \in Z} c_{j,k} \phi_{j,k}(t)$$

Si a esta ecuación la multiplicamos por $\phi_{j,k}(t)$,

$$\langle f(t), \phi_{j,k}(t) \rangle = c_{j,-\infty} \langle \phi_{j,-\infty}(t), \phi_{j,k}(t) \rangle + \dots + c_{j,k} \langle \phi_{j,k}(t), \phi_{j,k}(t) \rangle + \dots + c_{j,\infty} \langle \phi_{j,\infty}(t), \phi_{j,k}(t) \rangle$$

$$c_{j,k} = \langle f(t), \phi_{j,k}(t) \rangle = \int_{-\infty}^{\infty} |f(t) \phi_{j,k}(t)| dt$$

Dado que la función escala debe ser ortonormal, por lo que sólo uno de los productos puntos de la ecuación anterior es distinto de cero. Lo que nos queda:

$$c_{j,k} = \langle f(t), \phi_{j,k}(t) \rangle$$

1. Para cada $j \in Z$, $\{\phi_{j,k} | k \in Z\}$ forma una base ortonormal para el subespacio $V_j \subset L^2(R)$.
2. Los subespacios V_j están anidados, o sea

$$\forall j \in Z, \{V_j \subset V_{j+1}\}$$

3. La función $\phi(t)$ tiene soporte compacto. Esto es, existe un subconjunto del dominio de $\phi(t)$

donde esta no es cero, i.e. $\text{supp } \{\phi\} = \{x \in Z | \phi(x) \neq 0\}$.

5.2 Función Wavelet

Definimos

$$W_j = \overline{\text{Span}_{k \in \mathbb{Z}} \{\psi_{j,k}(t)\}}$$

como el complemento ortogonal de V_j en V_{j+1} , esto significa que todos los miembros de V_j son ortogonales a todos los miembros de W_j . Entonces se requiere que

$$\langle \phi_{j,k}(t), \psi_{j,k}(t) \rangle = \int_{-\infty}^{\infty} \phi_{j,k}(t) \psi_{j,k}(t) dt = 0 \quad j, k, l \in \mathbb{Z}$$

$$\psi_{j,k}(t) = 2^{\frac{j}{2}} \psi(2^j t - k) \quad j, k \in \mathbb{Z}$$

$$V_1 = V_0 \oplus W_0$$

$$f(t) = \sum_{k \in \mathbb{Z}} d_{j,k} \psi_{j,k}(t)$$

De este modo la función $\psi(t)$, que llamaremos *wavelet madre* se puede representar como

$$\psi(t) = \sum_{k \in \mathbb{Z}} \sqrt{2} h(n) \phi(2t - k)$$

6 Wavelets Haar

La transformada wavelet es un campo de investigación actual. Esto es debido a que el número de aplicaciones que se benefician de esta herramienta crece diariamente. En la literatura se pueden encontrar un sin número de aplicaciones de la transformada wavelet en distintas ramas: estadística, análisis de series temporales procesamiento de señales, procesamiento de imágenes, bases de datos, minería de datos.etc.

El término wavelet se define como una “pequeña onda” o función localizable en el tiempo. Vista desde una perspectiva del análisis o procesamiento de señales puede ser considerada como una herramienta matemática para la representación y segmentación de señales, análisis tiempo-frecuencia e implementación de algoritmos sencillos y rápidos desde el punto de vista computacional. Las características propias de la transformada wavelet nos otorgan la posibilidad de representar señales en diferentes niveles de resolución, representar en forma eficiente señales con variaciones de picos abruptos, así como analizar señales no

estacionarias. Nos permite conocer el contenido en frecuencia de una señal y cuándo estas componentes de frecuencia se encuentran presentes en la señal.

El análisis Wavelet representa el paso lógico siguiente a la STFT: una técnica mediante ventanas con regiones de tamaño variable. El análisis Wavelet permite el uso de intervalos grandes de tiempo en aquellos segmentos en los que se requiere mayor precisión en baja frecuencia, y regiones más pequeñas donde se requiere información en alta frecuencia.



6.1 Principios de Multi-Resolución

$$f(t) \in L^2(R)$$

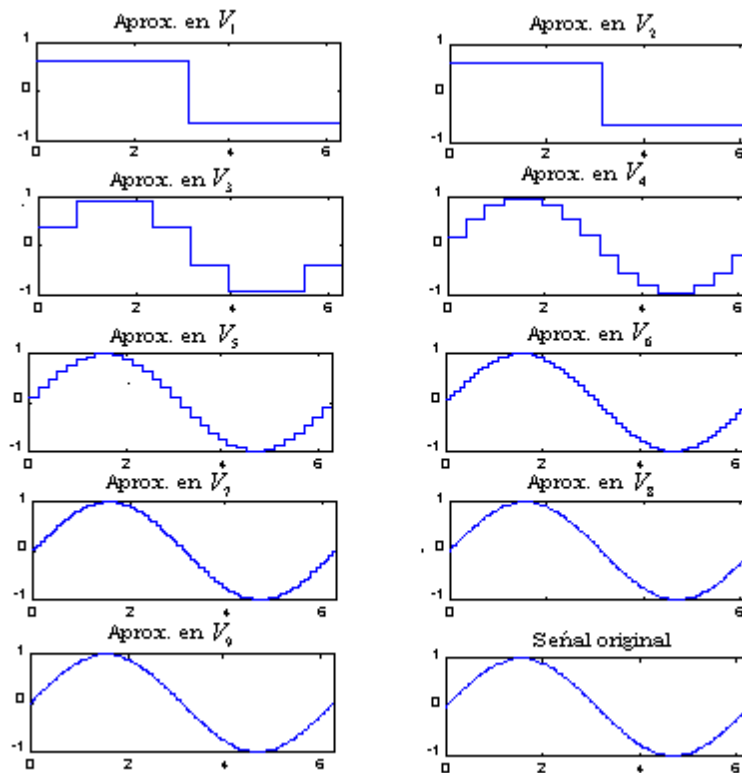
El análisis multi-resolución consiste básicamente en aproximar una función $f(t)$ en distintos niveles de resolución, lo que nos entrega una descomposición multi-escala de la forma:

$$f(t) = f_0(t) + \sum_{j \geq 0} g_j(t)$$

Un análisis multi-resolución requiere un anidamiento de los espacios generados por las funciones escala, de la forma:

$$\dots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots \subset L^2$$

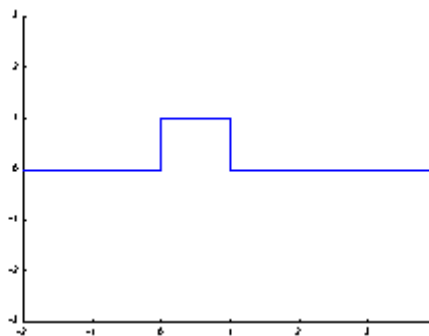
$$V_j \subset V_{j+1} \quad \forall j \in \mathbb{Z}$$



6.2 Función Haar Escala

Sea la función Haar definida de la siguiente manera:

$$\phi(t) = \begin{cases} 1 & \text{si } 0 \leq t < 1 \\ 0 & \text{el resto.} \end{cases}$$



$$f(t) = \sum_k c_k \phi_k(t) \quad k \in \mathbb{Z}$$

para cualquier $f(t) \in \nu_o \in L^2(\mathbb{R})$.

Cualquier función $f(t)$ que esté en ν puede ser representada por una combinación lineal del conjunto de funciones con sus respectivos coeficientes. Para un rápido cálculo de estos coeficientes es necesario que sea ortonormal, tiene que cumplir la siguiente condición:

$$\langle \phi_k(t) \cdot \phi_m(t) \rangle = \int \phi_k(t) \phi_m(t) dt = 0$$

$$\text{con } k \neq m$$

Ejemplo 1:

Representar la función $f(t) = t^2$ con $t \in [-3, 3]$, mediante la función de escalamiento $\phi_k(t)$ con $t \in [-3, 2]$.

Solución

Representaremos la función continua $f(t) = t^2$, como una combinación lineal de la forma

$$f(t) = c_{-3}\phi(t+3) + c_{-2}\phi(t+2) + c_{-1}\phi(t+1) + c_0\phi(t) + c_1\phi(t-1) + c_2\phi(t-2)$$

Multiplicamos por $\phi(t+3)$

$$t^2 = c_{-3}\phi(t+3) + c_{-2}\phi(t+2) + c_{-1}\phi(t+1) + c_0\phi(t) + c_1\phi(t-1) + c_2\phi(t-2) \odot \phi(t+3)$$

$$\langle t^2, \phi(t+3) \rangle = \langle c_{-3}\phi(t+3), \phi(t+3) \rangle + \dots + \langle c_2\phi(t-2), \phi(t+3) \rangle$$

Como la función $\phi_k(t)$ tiene que ser ortonormal, por lo que todos los productos serán cero menos uno, donde los límites de integración fueron calculados a partir de la función Haar:

$$\Rightarrow \langle t^2, \phi(t+3) \rangle = \langle c_{-3}\phi(t+3), \phi(t+3) \rangle$$

$$\Rightarrow \langle t^2, \phi(t+3) \rangle = \int_{-3}^{-3+1} c_{-3}\phi(t+3)\phi(t+3)dt$$

$$\Rightarrow \langle t^2, \phi(t+3) \rangle = c_{-3}[t]_{-3}^{-2}$$

$$\Rightarrow \langle t^2, \phi(t+3) \rangle = c_{-3}[-2+3]$$

$$\Rightarrow \langle t^2, \phi(t+3) \rangle = c_{-3}$$

$$\Rightarrow c_{-3} = \int_{-3}^3 t^2 \phi(t+3) dt$$

$$\Rightarrow c_{-3} = \int_{-3}^{-2} t^2 \phi(t+3) dt + \int_{-2}^3 t^2 \phi(t+3) dt$$

$$\Rightarrow c_{-3} = \int_{-3}^{-2} t^2 dt + \int_{-2}^3 t^2 * 0 dt$$

$$\Rightarrow c_{-3} = \left[\frac{t^3}{3} \right]_{-3}^{-2}$$

$$\Rightarrow c_{-3} = 6.\bar{3}$$

Estos pasos los seguimos para calcular todos los coeficientes, llegando a la conclusión de que cualquier coeficiente puede ser calculado de la siguiente manera:

$$c_k = \langle f(t), \phi_k(t) \rangle = \int_{t_1}^{t_2} f(t) dt$$

siempre que $f(t) \in L^2(R)$ y $\phi_k(t)$ sea ortonormal.

En determinadas ocasiones vemos que hay funciones en las que no se cumple la normalidad. En esas ocasiones necesitamos encontrar una constante que nos permita hacer estas funciones ortonormales.

$$\langle r_j \phi_{j,k}(t), r_j \phi_{j,k}(t) \rangle = 1$$

$$h(1) = \sqrt{2} [t]_0^{0.5} = \frac{\sqrt{2}}{2} = \frac{1}{\sqrt{2}}$$

$$h(2) = \sqrt{2} [t]_{0.5}^1 = \frac{\sqrt{2}}{2} = \frac{1}{\sqrt{2}}$$

En este apartado vamos a desarrollar la procedencia de las constantes $h(1)$ y $h(2)$. Debemos encontrar una constante que nos permita hacer estas funciones ortonormales,

Ejemplo 2

Representar la función $f(t) = t^2$ con $t \in [-3, 3]$, mediante la familia de funciones $\phi_{j,k}(t)$.

Solución

Para representar esta función trabajaremos en el espacio ν_1 , o lo que es lo mismo, $j = 1$.

$$t^2 = \sum_{k=-6}^5 c_k \phi(2t - k)$$

de manera análoga al ejemplo 1 (revisar desde la ecuación (6.12) a (6.19)) tenemos que

$$\langle t^2, \phi(2t + 6) \rangle = \langle c_{-6} \phi(2t + 6), \phi(2t + 6) \rangle$$

$$\langle c_{-6} \phi(2t + 6), \phi(2t + 6) \rangle = \int_{-3}^{-2.5} c_{-6} dt$$

donde los límites de integración están dados por (6.34)

$$\langle c_{-6} \phi(2t + 6), \phi(2t + 6) \rangle = \frac{c_{-6}}{2}$$

$$\frac{c_{-6}}{2} = \langle t^2, \phi(2t + 6) \rangle$$

$$c_{-6} = 2 \int_{-3}^{-2.5} t^2 dt$$

coeficientes	valores obtenidos
c_{-6}	7.58
c_{-5}	5.08
c_{-4}	3.08
c_{-3}	1.58
c_{-2}	0.58
c_{-1}	0.08
c_0	0.08
c_1	0.58
c_2	1.58
c_3	3.08
c_4	5.08
c_5	7.58

Del ejemplo anterior se puede observar que los coeficientes quedaban divididos por 2 lo que indica que las funciones de escalamiento de este subespacio no son ortonormales. Multipliquemos las funciones base por un número cualquiera r .

$$\langle r_j \phi_{j,k}(t), r_j \phi_{j,k}(t) \rangle = 1$$

$$\langle r_j \phi_{j,k}(t), r_j \phi_{j,k}(t) \rangle = \int_{t_1}^{t_2} r_j \phi_{j,k}(t) r_j \phi_{j,k}(t) dt = \frac{r_j^2}{2^j}$$

$$\frac{r_j^2}{2^j} = 1$$

$$r_j = 2^{\frac{j}{2}}$$

Con lo que obtenemos lo que denominaremos la constante de normalización. Con esta mejora definimos una nueva familia de funciones:

$$\phi_{j,k}(t) = 2^{\frac{j}{2}} \phi(2^j t - k) = \begin{cases} 2^{\frac{j}{2}} & \text{si } t_1 = \frac{k}{2^j} \leq t < \frac{1+k}{2^j} = t_2 \\ 0 & \text{caso contrario.} \end{cases}$$

$$\phi(t) = h(1)\sqrt{2}\phi(2t) + h(2)\sqrt{2}\phi(2t - 1)$$

$$h(1) = \sqrt{2} [t]_0^{0.5} = \frac{\sqrt{2}}{2} = \frac{1}{\sqrt{2}}$$

$$h(2) = \sqrt{2} [t]_{0.5}^1 = \frac{\sqrt{2}}{2} = \frac{1}{\sqrt{2}}$$

$$\phi(t) = \phi(2t) + \phi(2t - 1)$$

6.3 Función Haar Wavelet

Función Wavelet:

$$\psi(t) = \begin{cases} 1 & \text{si } 0 \leq t \leq \frac{1}{2} \\ -1 & \text{si } \frac{1}{2} \leq t \leq 1 \\ 0 & \text{caso contrario} \end{cases}$$

Al igual que el apartado anterior podemos seguir el mismo desarrollo para calcular los coeficientes Haar Wavelet. Podemos representar la función wavelet en el intervalo $[0,1]$ como combinación lineal de las siguientes funciones:

$$\psi(t) = h_1(1)\sqrt{2}\phi(2t) + h_1(2)\sqrt{2}\phi(2t-1)$$

$$h_1(1) = \sqrt{2} \int_0^{\frac{1}{2}} \psi(t) dt$$

$$h_1(2) = \sqrt{2} \int_{\frac{1}{2}}^1 \psi(t) dt \quad h_1(1) = -h_1(2) = \frac{1}{\sqrt{2}}$$

Observando que estos coeficientes nos permiten mantener la normalidad de la función. Una vez completado el desarrollo al igual que el apartado anterior veremos que ésta función a su vez necesitará otros coeficientes para mantener la normalidad.

Ejemplo 3

Aproximar la función $f(t) = t^2$ sobre el intervalo $[3, -3]$ en forma análoga a como se hizo con la función escalamiento.

Solución

La primera aproximación la haremos en W_0 de tal manera que las funciones wavelet de este espacio será de la forma $\psi_k(t) = \psi(t-k)$ con $k \in [-3, 2)$, y por lo tanto

$$f(t) = d_{-3}\psi(t+3) + d_{-2}\psi(t+2) + d_{-1}\psi(t+1) + d_0\psi(t) + d_1\psi(t-1) + d_2\psi(t-2)$$

Multiplicamos la ecuación anterior por

$$\psi(t+3)$$

$$t^2 = d_{-3}\psi(t+3) + d_{-2}\psi(t+2) + d_{-1}\psi(t+1) + d_0\psi(t) + d_1\psi(t-1) + d_2\psi(t-2) \odot \psi(t+3)$$

$$\langle t^2, \psi(t+3) \rangle = \langle d_{-3}\psi(t+3), \psi(t+3) \rangle + \dots + \langle d_2\psi(t-2), \psi(t+3) \rangle$$

Demostramos que es ortogonal ya que todos los productos son cero excepto uno de ellos:

$$\langle t^2, \psi(t+3) \rangle = \int_{-3}^{-3+1} d_{-3}\psi(t+3)\psi(t+3)dt$$

$$\int_{-3}^{-3+1} t^2\psi(t+3) = \int_{-3}^{-3+1} d_{-3}\psi(t+3)\psi(t+3)dt$$

que para $\psi(t+3) = 1$ nos queda

$$\int_{-3}^{-3+\frac{1}{2}} t^2 dt = \frac{d_{-3}^+}{2}$$

para $\psi(t+3) = -1$

$$\int_{-3+\frac{1}{2}}^{-2} -t^2 dt = \frac{d_{-3}^-}{2}$$

$$d_{-3}^+ = \frac{91}{12} = 7.58 \quad d_{-3}^- = -\frac{61}{12} = -5.08$$

Coeficientes	Valores Obtenidos
d_{-3}	2.5
d_{-2}	1.5
d_{-1}	0.5
d_0	-0.5
d_1	-1.5
d_2	-2.5

6.4 Transformada Wavelet Discreta (DWT)

La transformada Wavelet depende de dos variables, una encargada del escalamiento de la función wavelet a y otra encargada de la traslación de la función wavelet b . Mediante las variables de escalamiento y traslación la transformada Wavelet es capaz de hacer un análisis tiempo - frecuencia con una resolución variable, es decir, utiliza ventanas de diferente ancho durante el análisis de la señal. La función escala ϕ es la encargada de analizar el comportamiento general de la señal, mientras que la función wavelet ψ se encarga de analizar el comportamiento del detalle de la señal.

$$f(t) = \sum_{k=0}^{2^{j_0}-1} c_{j_0,k} \phi_{j_0,k}(t) + \sum_{j=j_0}^{N-1} \sum_{k=0}^{2^j-1} d_{jk} \psi_{jk}(t) \quad j, k \in \mathbb{Z}^+$$

Donde c_j son los coeficientes de escala, d_j son los coeficientes wavelet y j_0 nos entrega el espacio inicial V_{j_0} que será el espacio de menor resolución, y dependiendo de este j_0 es que el resto de los índices seguirán corriendo.

A continuación para ver el gran volumen de cálculos que implica realizar la transformada discreta wavelet vamos a realizar un ejemplo de cómo se calcularía.

Tomemos la función seno (t) en los siguientes puntos:

$$f(t) = [0, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}}, -1, -\frac{1}{\sqrt{2}}] \in L^2(\mathbb{R}) \quad t \in [0, 2\pi]$$

Tomamos la definición DWT

$$f(t) = \sum_{k=0}^{2^{j_0}-1} c_{j_0,k} \phi_{j_0,k}(t) + \sum_{j=j_0}^{N-1} \sum_{k=0}^{2^j-1} d_{jk} \psi_{jk}(t) \quad j, k \in \mathbb{Z}^+$$

Tomamos la definición Wavelet Haar

$$\psi_{jk}(t) = 2^{(j/2)} \psi(2^j t - k)$$

$$\phi_{jk}(t) = 2^{(j/2)} \phi(2^j t - k)$$

Sustituimos la definición de Wavelet Haar en DWT y desarrollamos la sumatorio para $j_0=0$.

$$g(t) = c_{00} \phi(t) + d_{00} \psi(t) + d_{10} \sqrt{2} \psi(2t) + d_{11} \sqrt{2} \psi(2t - 1) + d_{20} 2 \psi(4t) + \dots$$

- $\phi(t) = 1 \text{ si } 0 \leq t < 1$
- $\psi(t) = \begin{cases} 1 & \text{si } 0 \leq t < 0.5 \\ -1 & \text{si } 0.5 \leq t < 1 \end{cases}$
- $\psi(2t) = \begin{cases} 1 & \text{si } 0 \leq 2t < 0.5 \\ -1 & \text{si } 0.5 \leq 2t < 1 \end{cases} \Rightarrow \psi(2t) = \begin{cases} 1 & \text{si } 0 \leq t < 0.25 \\ -1 & \text{si } 0.25 \leq t < 0.5 \end{cases}$
- $\psi(2t - 1) = \begin{cases} 1 & \text{si } 0.5 \leq t < 0.75 \\ -1 & \text{si } 0.75 \leq t < 1 \end{cases}$
- $\psi(4t) = \begin{cases} 1 & \text{si } 0 \leq t < 0.125 \\ -1 & \text{si } 0.125 \leq t < 0.25 \end{cases}$
- $\psi(4t - 1) = \begin{cases} 1 & \text{si } 0.25 \leq t < 0.375 \\ -1 & \text{si } 0.375 \leq t < 0.5 \end{cases}$
- $\psi(4t - 2) = \begin{cases} 1 & \text{si } 0.5 \leq t < 0.625 \\ -1 & \text{si } 0.625 \leq t < 0.75 \end{cases}$
- $\psi(4t - 3) = \begin{cases} 1 & \text{si } 0.75 \leq t < 0.875 \\ -1 & \text{si } 0.875 \leq t < 1 \end{cases}$

$$\begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 1 \\ \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \\ -1 \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = 2^{\frac{j_0}{2}} \begin{bmatrix} 1 & 1 & \sqrt{2} & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & \sqrt{2} & 0 & -2 & 0 & 0 & 0 \\ 1 & 1 & -\sqrt{2} & 0 & 0 & 2 & 0 & 0 \\ 1 & 1 & -\sqrt{2} & 0 & 0 & -2 & 0 & 0 \\ 1 & -1 & 0 & \sqrt{2} & 0 & 0 & 2 & 0 \\ 1 & -1 & 0 & \sqrt{2} & 0 & 0 & -2 & 0 \\ 1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & 2 \\ 1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} C_{00} \\ d_{00} \\ d_{10} \\ d_{11} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix}$$

$$\begin{bmatrix} c_{00} \\ d_{00} \\ d_{10} \\ d_{11} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix} = 2^{\frac{-j_0}{2}} A^{-1} \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 1 \\ \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \\ -1 \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} c_{00} \\ d_{00} \\ d_{10} \\ d_{11} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.6036 \\ -0.1768 \\ 0.1768 \\ -0.1768 \\ 0.0732 \\ 0.1768 \\ -0.0732 \end{bmatrix}$$

$$\begin{bmatrix} c_{10} \\ c_{11} \\ d_{10} \\ d_{11} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0.4268 \\ -0.4268 \\ -0.1768 \\ 0.1768 \\ -0.1768 \\ 0.0732 \\ 0.1768 \\ -0.0732 \end{bmatrix} \begin{bmatrix} c_{20} \\ c_{21} \\ c_{22} \\ c_{23} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix} = \frac{1}{\sqrt{2}\sqrt{2}} \begin{bmatrix} 0.1768 \\ 0.4268 \\ -0.1768 \\ -0.4268 \\ -0.1768 \\ 0.0732 \\ 0.1768 \\ -0.0732 \end{bmatrix}$$

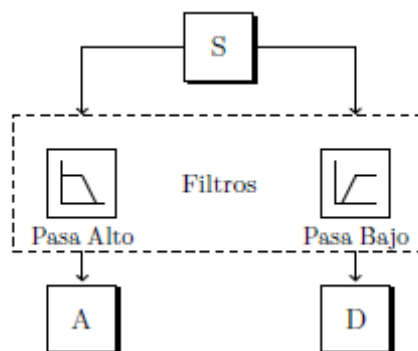
$$\begin{bmatrix} c_{30} \\ c_{31} \\ c_{32} \\ c_{33} \\ c_{34} \\ c_{35} \\ c_{36} \\ c_{37} \end{bmatrix} = \frac{1}{\sqrt{2}\sqrt{2}\sqrt{2}} \begin{bmatrix} 0 \\ 2 \\ \sqrt{8} \\ 2 \\ 0 \\ -2 \\ -\sqrt{8} \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 1 \\ \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \\ -1 \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

6.5 TRANSFORMADA RÁPIDA WAVELET (FWT) Y BANCO DE FILTROS

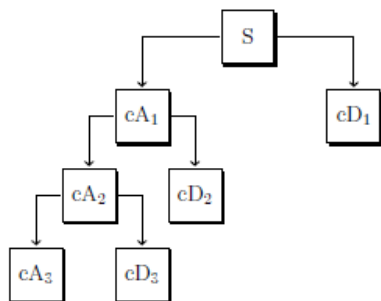
En la mayoría de las señales son las componentes de baja frecuencia las que le otorgan a la señal la mayor parte de su información. Mientras que las componentes de alta frecuencia se encargan de incorporar características más particulares. Es por ello que se subdividen las Componentes de una señal en dos categorías:

Aproximaciones (baja frecuencia)

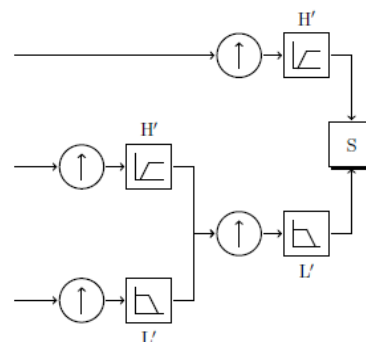
Detalles (alta frecuencia)



Descomposición de la Señal



Reconstrucción de la Señal



Algebraicamente, el objetivo de aplicar la DWT a un vector es obtener un vector transformado que tiene en la mitad, conocida como parte alta, la misma información de alta frecuencia que el vector original y en otra mitad, conocida como parte baja, la información de baja frecuencia. El éxito de la DWT en el procesamiento de señales reside en el hecho de que usualmente la mayor parte de la información de alta frecuencia es relativamente pequeña y puede descartarse, permitiendo así una compresión eficiente de los datos.

Representación Algebraica de la DWT-1D

Baja frecuencia filtro G

Altas frecuencias filtro H

$$W = \begin{pmatrix} G \\ H \end{pmatrix} \quad (\text{Figura 1})$$

$$H = \begin{pmatrix} h_1 & h_2 & \cdots & h_L & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & h_1 & h_2 & \cdots & h_L & 0 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \ddots & & \ddots & \ddots & & \vdots & \\ h_3 & h_4 & \cdots & h_L & 0 & \cdots & & 0 & h_1 & h_2 \end{pmatrix}_{\frac{n}{2} \times n}$$

$$G = \begin{pmatrix} g_1 & g_2 & \cdots & g_L & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & g_1 & g_2 & \cdots & g_L & 0 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \ddots & & \ddots & \ddots & & \vdots & \\ g_3 & g_4 & \cdots & g_L & 0 & \cdots & & 0 & g_1 & g_2 \end{pmatrix}_{\frac{n}{2} \times n}$$

Al multiplicar la matriz de la figura 1 por un vector v :

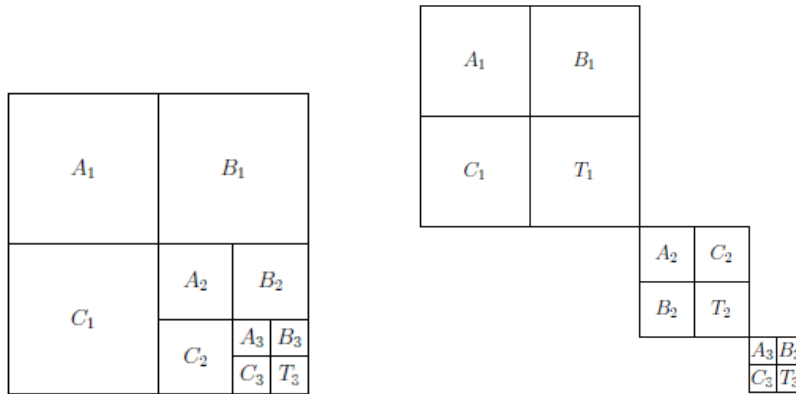
$$Wv = \begin{pmatrix} Hv \\ Gv \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_{n/2} \\ s_1 \\ \vdots \\ s_{n/2} \end{pmatrix}$$

donde d_i son los componentes de alta frecuencia (componentes de detalle) y s_i son los componentes de baja frecuencia (componentes de aproximación).

Representación Algebraica de la DWT - 2D

$$\tilde{T}_0 = WT_0W^T = \begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix} \quad T_1 = GT_0G^T, \quad A_1 = HT_0H^T, \quad B_1 = HT_0G^T, \quad C_1 = GT_0H^T$$

Matriz descompuesta en 3 niveles de la transformada wavelet



Transformada rápida de k niveles wavelet de un vector de longitud N

```

n := N
para l := 1 to k hacer
  Inicializar a w como un vector nulo de longitud n
  para i := 0 to  $\frac{n}{2} - 1$  hacer
    para j := 0 to m - 1 hacer
       $w_i = w_i + h_j v_{(2i+j)_n}$ 
       $w_{i+\frac{n}{2}} = w_i + g_j v_{(2i+j)_n}$ 
    fin para
  fin para
   $v_{0:n-1} = w$ 
   $n = \frac{n}{2}$ 
fin para

```

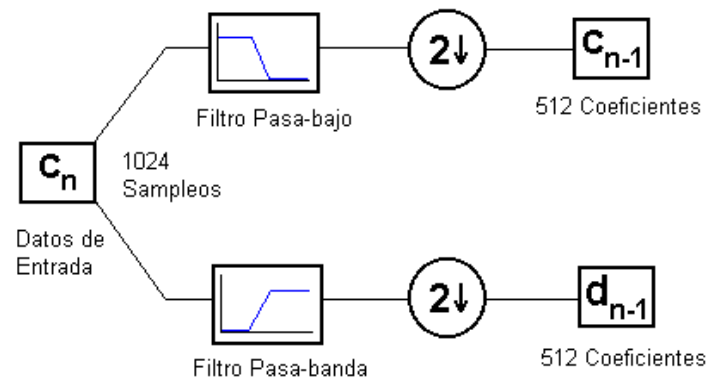
Transformada rápida inversa wavelet de k niveles de un vector de Longitud N

```

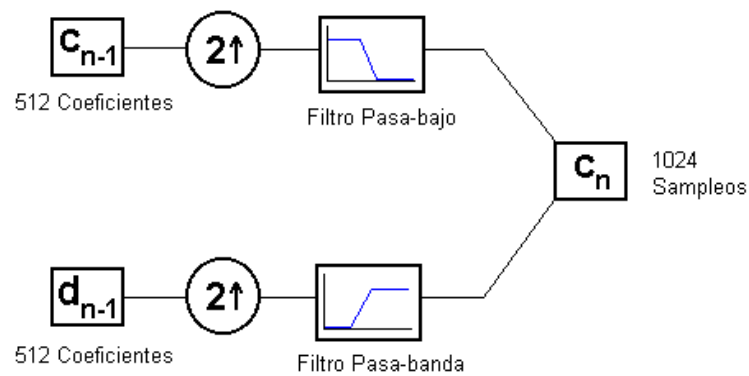
n :=  $\frac{N}{2^{k-1}}$ 
para l := 1 to k hacer
  Inicializar a w como un vector nulo de longitud n
  para i := 0 to  $\frac{n}{2} - 1$  hacer
    para j := 0 to m - 1 hacer
       $w_{(2i+j)_n} = w_{(2i+j)_n} + h_j v_i$ 
       $w_{(2i+j)_n} = w_{(2i+j)_n} + g_j v_{i+\frac{n}{2}}$ 
    fin para
  fin para
   $v_{0:n-1} = w$ 
   $n = 2n$ 
fin para

```


Proceso de Descomposición de la Señal



Proceso de Reconstrucción de la Señal



Ejemplo Descomposición - Reconstrucción

Conocimientos previos

$$h(0) = h(1) = \frac{1}{\sqrt{2}}$$

$$h_1(0) = \frac{1}{\sqrt{2}} \quad h_1(1) = -\frac{1}{\sqrt{2}}$$

Coeficientes Descomposición

$$c_{J-1,k} = \frac{1}{\sqrt{2}}(c_{J,2k} + c_{J,2k+1})$$

$$d_{J-1,k} = \frac{1}{\sqrt{2}}(c_{J,2k} - c_{J,2k+1})$$

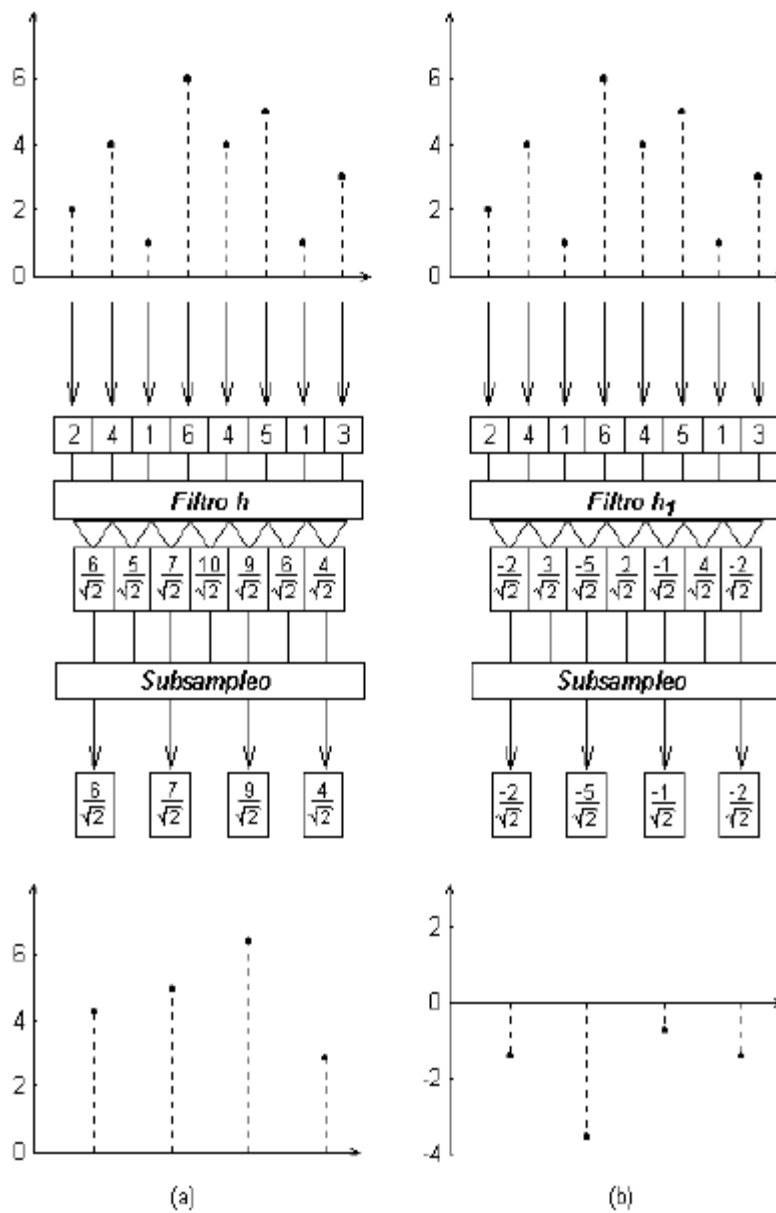
Coeficientes Reconstrucción:

$$c_{J,2k} = \frac{1}{\sqrt{2}}(c_{J-1,k} + d_{J-1,k})$$

$$c_{J,2k-1} = \frac{1}{\sqrt{2}}(c_{J-1,k} - d_{J-1,k})$$

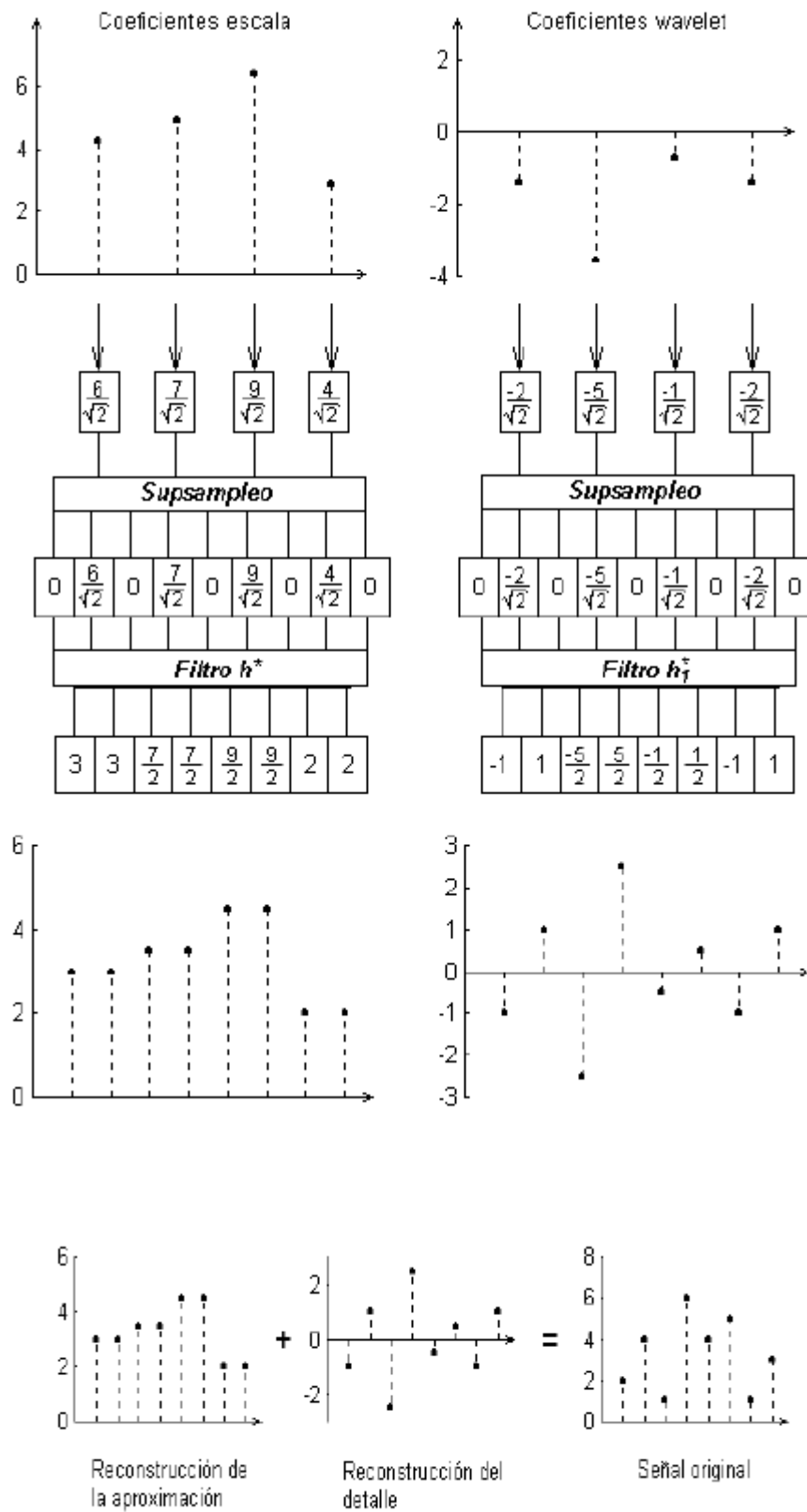
Datos de Entrada

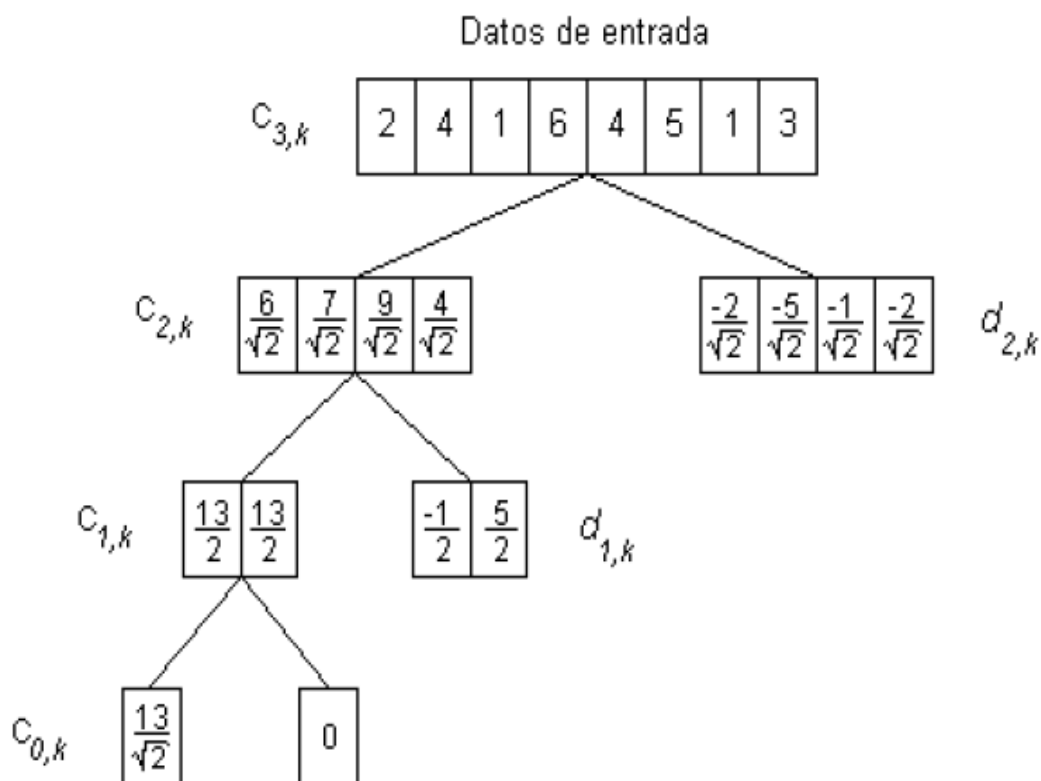
Supongamos una señal con una longitud $N=8$ definida como $Y = \{2, 4, 1, 6, 4, 5, 1, 3\}$.



(a) Coeficientes de aproximación.

(b) Coeficientes Wavelet





$C_{0,0}$	$d_{0,0}$	$d_{1,0}$	$d_{1,1}$	$d_{2,0}$	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Vector - DWT

$$\left\{ \frac{13}{\sqrt{2}}, 0, -\frac{1}{2}, \frac{5}{2}, -\frac{2}{\sqrt{2}}, -\frac{5}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, -\frac{2}{\sqrt{2}} \right\}$$

7 Aplicaciones

7.1 Aplicaciones de la DWT en la solución de sistemas de ecuaciones lineales

Existen varias líneas de investigación, orientadas a la solución eficiente de sistemas de ecuaciones lineales (Beylkin, 1991; Zhou, 2000 y Lega, 1995). El algoritmo de principal interés es el Precondicionador Multinivel propuesto por Chan (2003). En este trabajo se usa la descomposición (1) producida por la DWT en combinación con el método Complemento de *Schur* y con métodos iterativos estándares como GMRES y Richardson (Chen, 2003) para producir un algoritmo multinivel.

Ax=B Algoritmo Wavelet-Schur

$$\bar{A}_{i,j} = \begin{cases} A_{i,j} & \text{si } |i - k| < \mu \\ 0, & \text{en otro caso} \end{cases}$$

$$\bar{T}_0 = \begin{pmatrix} \bar{A}_1 & \bar{B}_1 \\ \bar{C}_1 & T_1 \end{pmatrix}$$

$$\begin{pmatrix} \bar{A}_1 & \bar{B}_1 \\ \bar{C}_1 & T_1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

ALGORITMO

Entradas:

A: matriz del sistema; b: Vector de términos independientes

x: Aproximación inicial de la solución; μ_j : cantidad máxima de iteraciones en el nivel j, tol: nivel de tolerancia de error de la soluciónj: nivel *wavelet* Actual. T_j : matriz preconditionador en el nivel j, inicialmente es nula*Wavelet-Schur*Si No estamos en el nivel más grueso ($j < l$)

{

Si es el nivel más fino ($j == 0$) entonces $T_j = A$ (1) Aplicar **FWT**(1 nivel) a $T_j x_j = b_j$ para obtener $\bar{T}_j \tilde{x}_j = \tilde{b}_j$

$$\text{Obtener preconditionador } \bar{T}_j = \begin{bmatrix} \bar{A}_{j+1} & \bar{B}_{j+1} \\ \bar{C}_{j+1} & T_{j+1} \end{bmatrix}$$

Aplicar (GMRES) con μ_j pasos para resolver $\bar{T}_j \tilde{x}_j = \tilde{b}_j$ usando \bar{T}_j como preconditionador

INICIO //GMRES

...

En cada paso resolver $\bar{T}_j y_j = r_j$ (Sistema preconditionador) en complemento de Schur

$$(2) \quad \bar{A}_{j+1} z_1 = r_0^{(1)},$$

$$z_2 = r_0^{(2)} - \bar{C}_{j+1} z_1,$$

$$(3) \quad \text{Wavelet-Schur}(T_{j+1} - \bar{C}_{j+1} \bar{A}_{j+1}^{-1} \bar{B}_{j+1}, Z_2, y_0^{(2)}, \mu_j, \text{tol}, l, j+1, T_j)$$

$$(4) \quad y_0^{(1)} = z_1 - \bar{A}_{j+1}^{-1} \bar{B}_{j+1} y_0^{(2)}$$

...

FIN (Schultz)

(5) Aplicar un nivel de **FWT inversa** a \tilde{x}_j para obtener x_j

Sino

(6) **Resolver** por un método directo $Ax = b$

7.2 Compresión

Existen tres pasos principales a la hora de aplicar la compresión de imágenes con Wavelets.

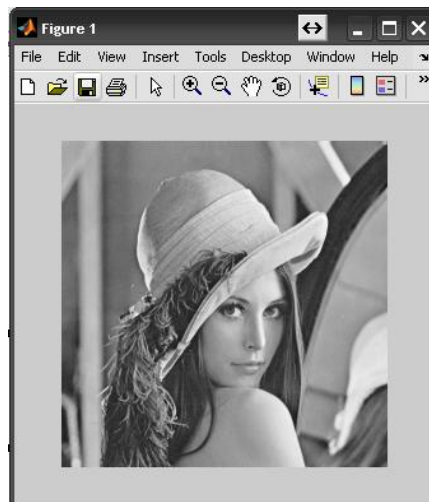
1. Proyectar la señal original a un subespacio multiresolución V_n con un n lo suficientemente grande.
2. Aplicar el algoritmo de descomposición de Wavelets.
3. Establecer un esquema de cuantización.

7.2.1 Compresión de Imágenes con MatLab 7.0

A continuación vamos a incluir la compresión de Imágenes con MatLab 7.0. Vamos a realizar la transformada de Wavelet de primer nivel de una imagen bitmap con una resolución de 256x256 en blanco y negro.

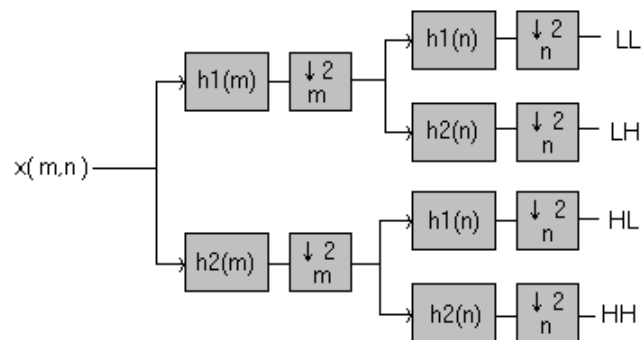
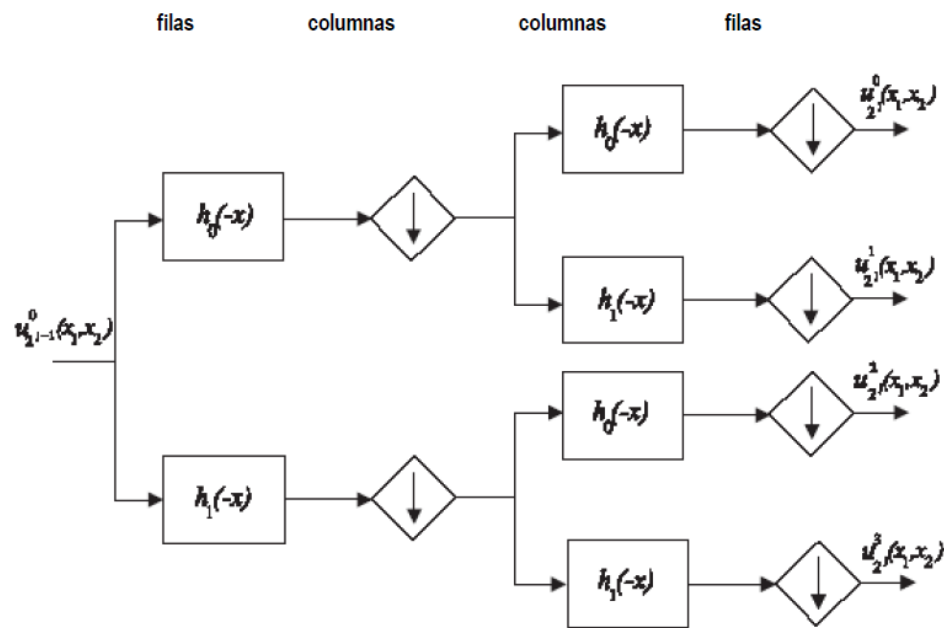
CARGA Y VISUALIZACIÓN DE LA IMAGEN A PROCESAR EN MATLAB:

```
[nom,path]=uigetfile('*.bmp');
a=imread([path,nom]);
b=double(a);
b=b/max(max(b));
imshow(b);
```



PRIMERA ETAPA DE LA TRANSFORMADA DIRECTA.

Recordemos que el diagrama de bloques para la descomposición en subbandas de la imagen es:



Los filtros base $h_1(n)$ y $h_2(n)$ que pueden interpretarse como un filtro paso bajo y uno paso alto respectivamente. Los filtros h_1 y h_2 pueden ser cualesquiera que cumplan las condiciones exigidas por la transformada wavelet y pueden expresarse en la dirección m – horizontal de la siguiente manera:

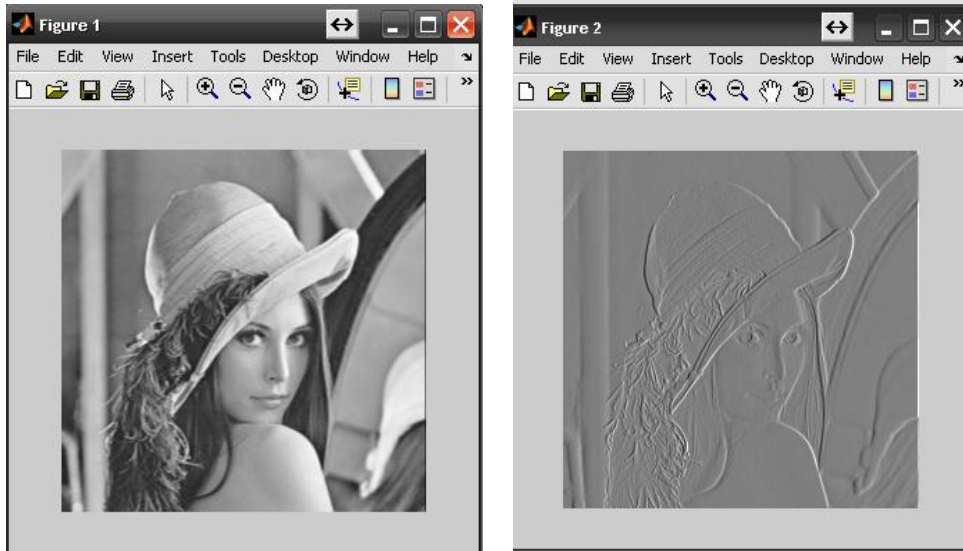
$$h_1 = [1/\sqrt{2}, 1/\sqrt{2}];$$

$$h_2 = [1/\sqrt{2}, -1/\sqrt{2}];$$

$$bh_1 = \text{filter2}(h_1, b);$$

$$bh_2 = \text{filter2}(h_2, b);$$

Obsérvese que las dos imágenes son una versión filtrada de la imagen original. En el primer caso, se ha realizado un suavizado en la dirección horizontal mientras en el segundo se han enfatizado los contornos verticales (filtro paso alto en la dirección horizontal).



Realizamos un diezmado de la imagen sin que esta pierda información, se reduce la resolución, pasando a obtener imágenes de 256x128 píxeles. La imagen de baja frecuencia se normaliza debido a que el filtro paso bajo tiene una ganancia distinta de la unidad. La descomposición de la imagen original en estas dos subimágenes permite reducir la frecuencia de muestreo en la dirección horizontal sin que se produzca, como veremos posteriormente, pérdida de información.

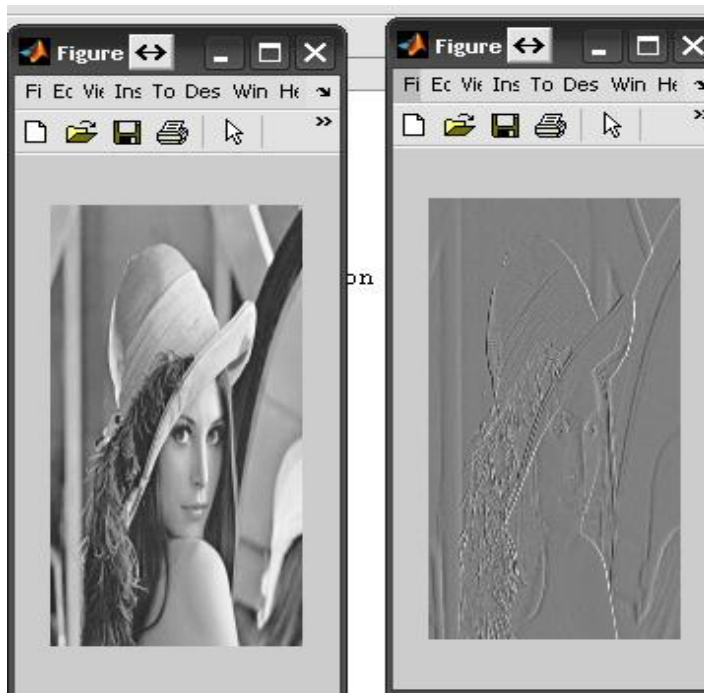
```
bh1m=bh1(:,1:2:256);
```

```
bh2m=bh2(:,1:2:256);
```

```
figure(1); imshow(bh1m/max(max(bh1m)));
```

```
figure(2); imshow(bh2m+0.5);
```

Nótese que al realizar el diezmado en la dirección horizontal se reduce la resolución, pasando a obtener imágenes de 256x128 píxeles.



Aplicaremos ahora los filtrados en la dirección vertical a cada una de las subimágenes obtenidas:

```
bh1mh1=filter2(h1',bh1m);
```

```
bh1mh2=filter2(h2',bh1m);
```

```
bh2mh1=filter2(h1',bh2m);
```

```
bh2mh2=filter2(h2',bh2m);
```

Ahora podemos aplicar el diezmado vertical a estas imágenes:

```
bLL=bh1mh1(1:2:256,:);
```

```
bLH=bh1mh2(1:2:256,:);
```

```
bHL=bh2mh1(1:2:256,:);
```

```
bHH=bh2mh2(1:2:256,:);
```

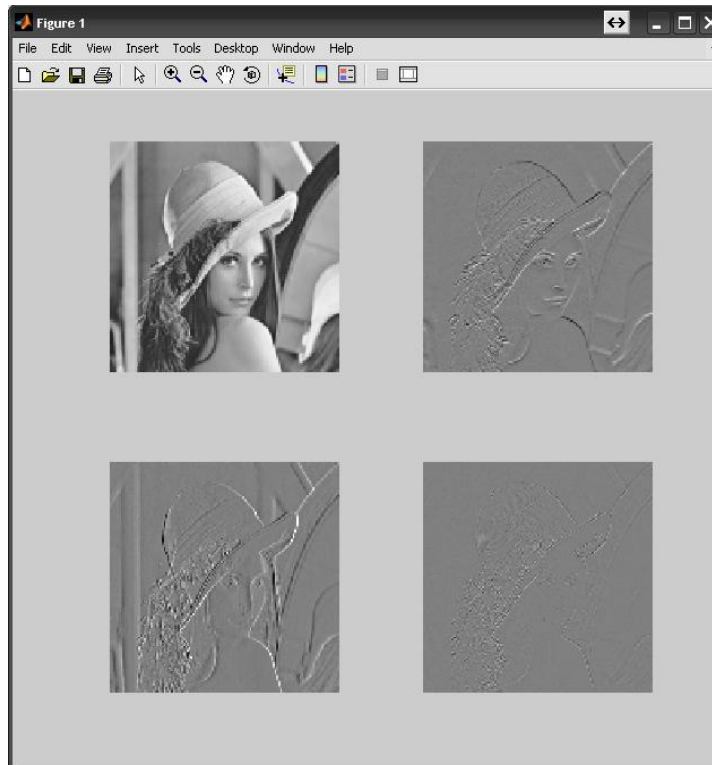
Estas imágenes se corresponden con las salidas del diagrama de bloques que representa el análisis por bandas de la transformada wavelet. Con el diezmado en las direcciones verticales y horizontales se han obtenido 4 imágenes de 128x128 puntos que representan la imagen original de 256x256. Esto, en principio, no supone ninguna compresión del número total de píxeles. No obstante, veremos más adelante que las imágenes con alguna componente de alta frecuencia pueden codificarse de forma mucho más eficiente sin que se produzcan pérdidas de información. Las imágenes resultantes de la descomposición en bandas son:

```
figure(1); subplot(221);imshow(bLL/max(max(bLL)));
```

```
subplot(222);imshow(bLH+0.5);
```

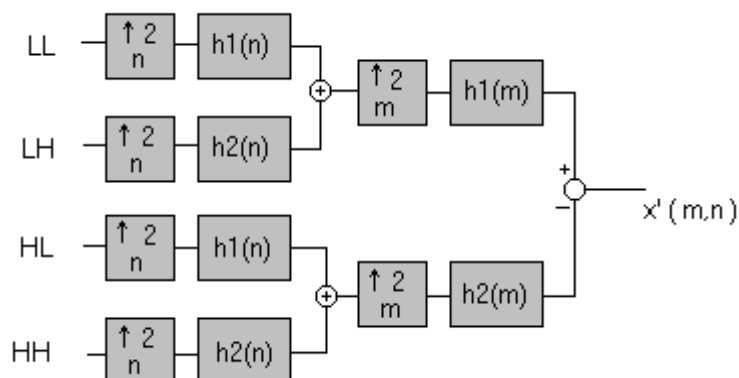
```
subplot(223);imshow(bHL+0.5);
```

```
subplot(224);imshow(bHH+0.5);
```



LA TRANSFORMACIÓN INVERSA.

Las imágenes bLL, bLH, bHL y bHH constituyen una representación alternativa de la imagen original b. Podemos reconstruir esta imagen original a partir de las anteriores sin más que aplicar las operaciones de la transformación en orden inverso. El diagrama de bloques para restaurar la imagen original es:



Seguiremos este esquema para obtener la imagen reconstruida *br*. En primer lugar intercalamos ceros en la dirección vertical de las imágenes de partida, obteniendo imágenes de 256x128 píxeles.

```
bLLin=zeros(256,128);bLLin(1:2:256,1:128)=bLL;
```

```
bLHin=zeros(256,128);bLHin(1:2:256,1:128)=bLH;
```

```
bHLin=zeros(256,128);bHLin(1:2:256,1:128)=bHL;
```

```
bHHin=zeros(256,128);bHHin(1:2:256,1:128)=bHH;
```

Podemos visualizar una de las imágenes resultantes para comprobar la correcta intercalación de los ceros:

```
imshow(bLLin/max(max(bLLin)))
```

Ahora filtramos las imágenes resultantes mediante *h1* y *h2* tal y como se indica en el diagrama de bloques de la figura anterior:

```
h2=[-1/sqrt(2),1/sqrt(2)];
```

(Nota: El filtro reconstructor paso alto tiene los coeficientes cambiados de orden)

```
bL1=filter2(h1',bLLin)+filter2(h2',bLHin);
```

```
bH1=filter2(h1',bHLin)+filter2(h2',bHHin);
```

Intercalamos los zeros en la dirección horizontal a cada una de las imágenes:

```
bLin=zeros(256,256); bLin(1:256,1:2:256)=bL1;
```

```
bHin=zeros(256,256); bHin(1:256,1:2:256)=bH1;
```

Y nuevamente filtramos y combinamos las imágenes para obtener la imagen reconstruida:

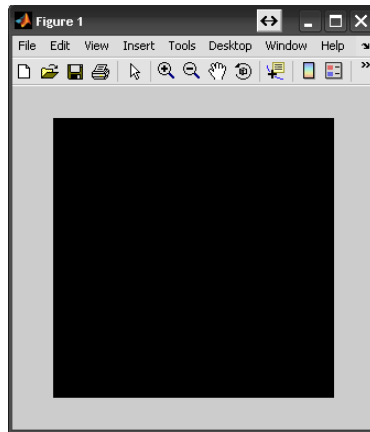
```
br = filter2(h1,bLin)+filter2(h2,bHin);
```

La imagen resultante resulta:

```
imshow(br/max(max(br)))
```

```
imshow(abs(b(2:256,2:256)-br(1:255,1:255)))
```

Lo cual indica que ambas imágenes son idénticas salvo el posible desplazamiento de un píxel, que se debe a los retardos introducidos por los filtros y que se puede estimar a priori.



Es importante comprobar que el número de elementos de imagen en las dos representaciones son idénticos. En efecto la imagen original tiene un total de $256 \times 256 = 65.536$ píxeles mientras que cada una de las imágenes transformadas tienen $128 \times 128 = 16.384$ píxeles que resulta en un total de $4 \times 16.384 = 65.536$ píxeles.

Hasta aquí, la representación de la imagen mediante las 4 componentes de su transformada wavelet no representa ninguna pérdida de información, pudiendo recuperar de forma exacta la imagen original a partir de sus transformadas. Otra ventaja de la transformada wavelet es que las componentes de alta frecuencia pueden almacenarse de forma aproximada sin que ello produzca pérdidas de calidad aparente en la imagen decodificada. En efecto, ahora aproximaremos las imágenes bLH, bHL y bHH en pasos de 0.1 unidades y veremos que no se observan grandes pérdidas en la reconstrucción de la imagen original. Para aproximar las componentes de alta frecuencia en pasos de 0.1 unidades utilizamos las siguientes instrucciones:

```
bLHa = round(bLH*10)/10;
```

```
bHLa = round(bHL*10)/10;
```

```
bHHa = round(bHH*10)/10;
```

Para ver como esta reducción en el número de niveles ha afecta a la calidad de la imagen vamos a repetir el proceso de reconstrucción de la imagen partiendo de bLL, bLHa, bHLa y bHHa. El proceso es idéntico al que se ha utilizado en el apartado de reconstrucción exacta de la imagen original a partir de la transformada wavelet.

```
bLLin=zeros(256,128);bLLin(1:2:256,1:128)=bLL;
```

```
bLHin=zeros(256,128);bLHin(1:2:256,1:128)=bLHa;
```

```
bHLin=zeros(256,128);bHLin(1:2:256,1:128)=bHLa;
```

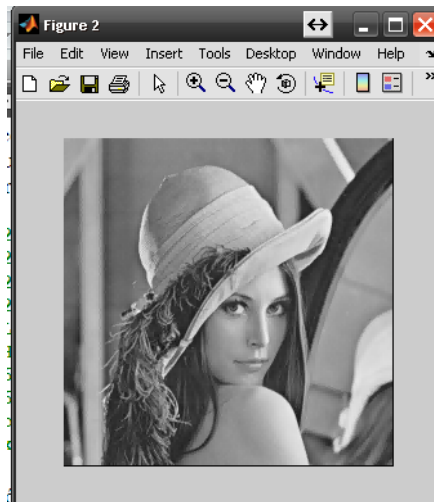
```
bHHin=zeros(256,128);bHHin(1:2:256,1:128)=bHHa;
```

```
bL1=filter2(h1',bLLin)+filter2(h2',bLHin);
```

```

bH1=filter2(h1',bHLin)+filter2(h2',bHHin);
bLin=zeros(256,256); bLin(1:256,1:2:256)=bL1;
bHin=zeros(256,256); bHin(1:256,1:2:256)=bH1;
bra = filter2(h1,bLin)+filter2(h2,bHin);
imshow(bra/max(max(bra)));

```

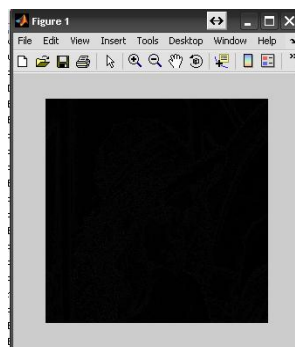


Ahora, la reconstrucción sólo es aproximada, pero los efectos de esta aproximación son poco visibles. Para comprobar la aproximación podemos comparar la imagen br, con la imagen original mediante el valor absoluto de la diferencia.

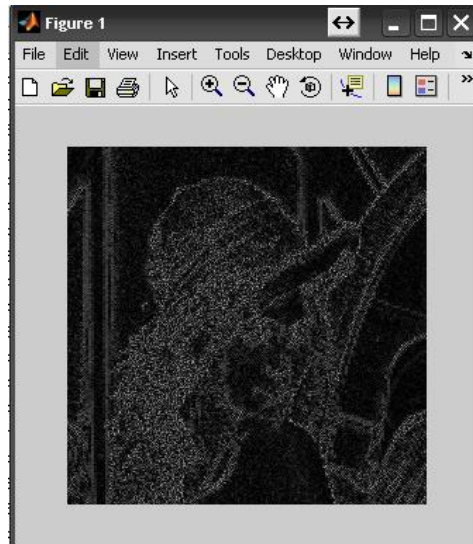
```

imshow(abs(b(2:256,2:256)-bra(1:255,1:255)))

```

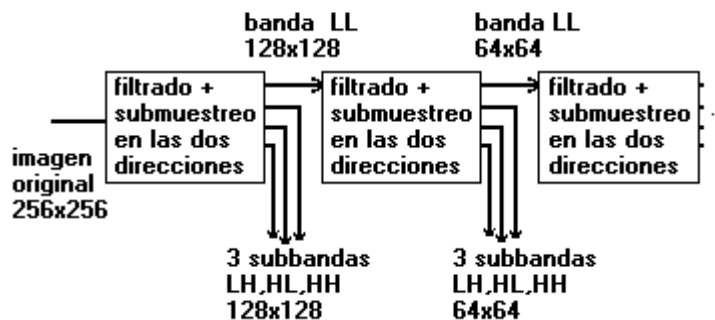


Las diferencias sólo son aparentes cuando aumentamos la escala de la imagen diferencia en un factor 10: `imshow(10*abs(b(2:256,2:256)-bra(1:255,1:255)))`



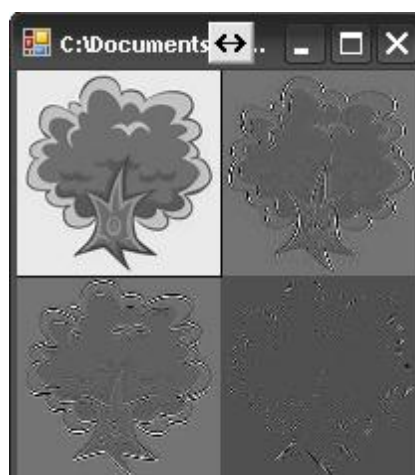
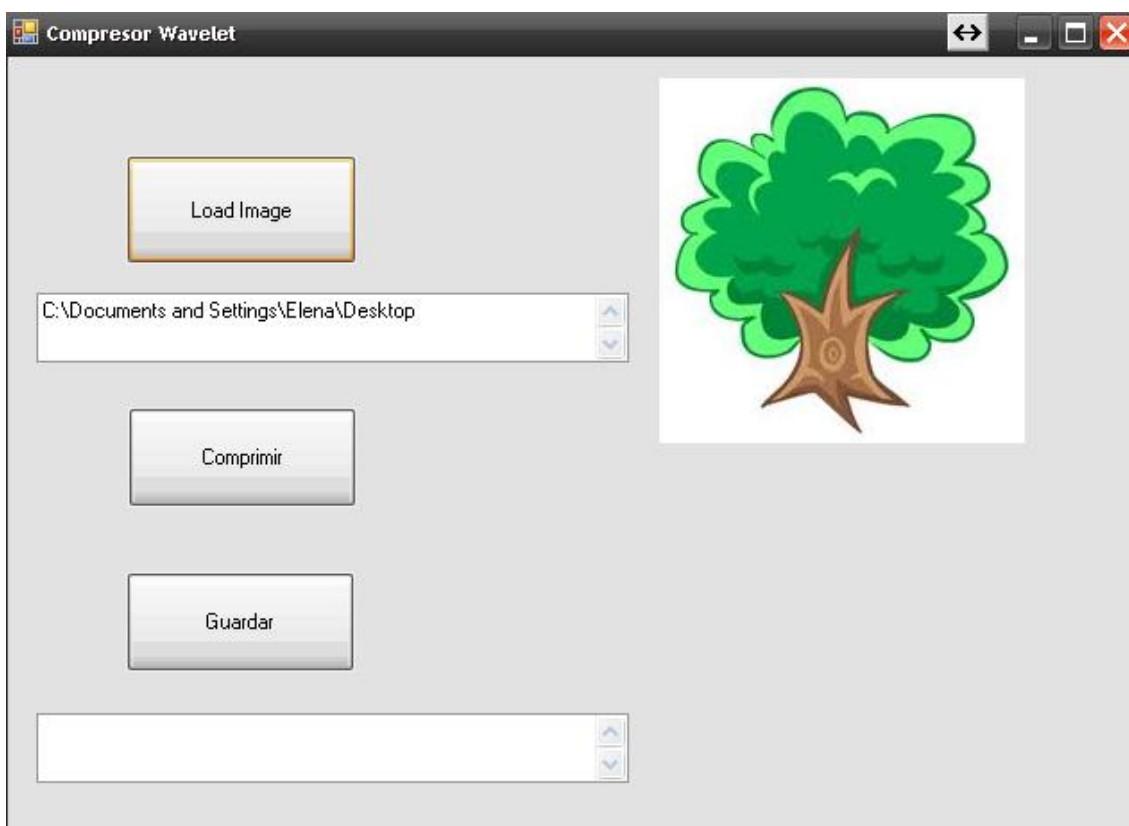
En resumen, hemos comprobado que las componentes de alta frecuencia de la transformada wavelet pueden codificarse de forma mucho más burda que los componentes de baja frecuencia sin que aparezcan pérdidas aparentes en la calidad de la imagen. Obsérvese que la mayor parte de los coeficientes de bLHa, bHLA y bHHa toman el valor cero, de modo que, en la práctica resulta conveniente codificar únicamente la posición y el valor de los elementos distintos de cero con lo que puede aumentarse todavía más el factor de compresión.

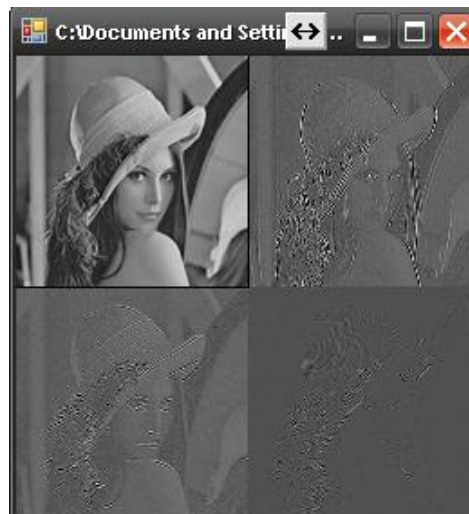
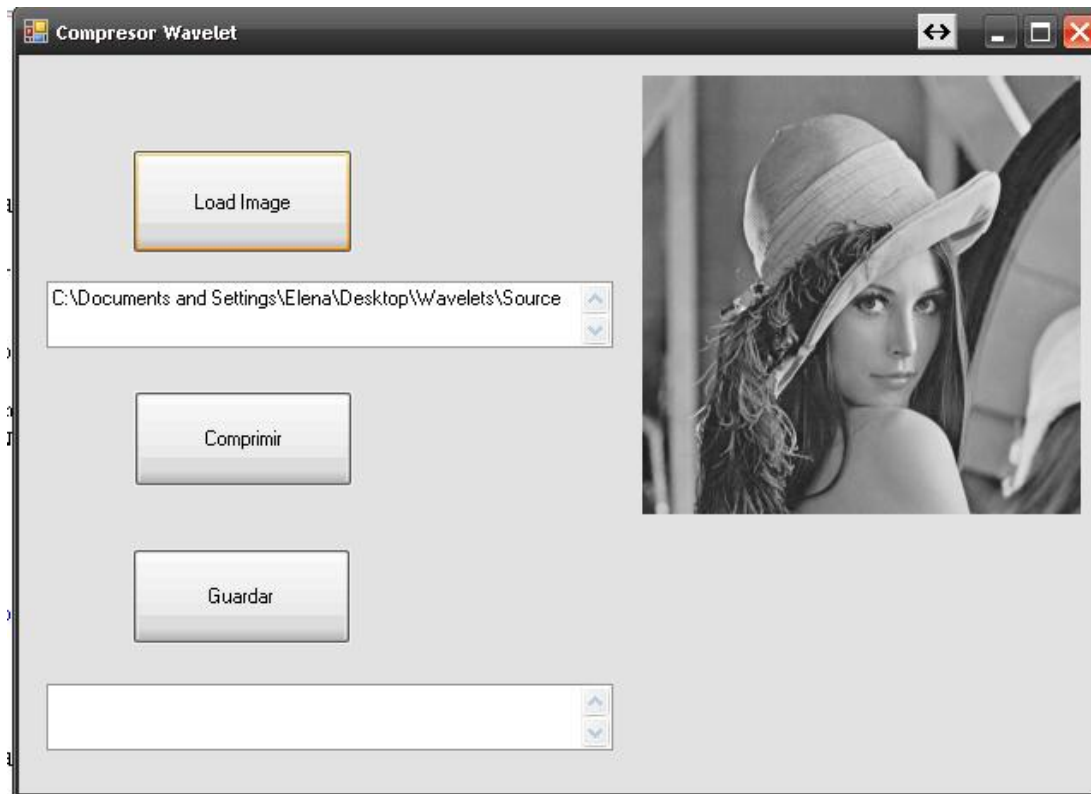
Hasta ahora, queda claro que hemos obtenido una importante ventaja en la codificación de la imagen mediante la representación que nos proporciona la transformada wavelet. No obstante, la imagen de baja frecuencia bLL, aún exige un elevado número de bits para su correcta codificación. La consecuencia lógica de estos resultados es evidente: podemos obtener considerables mejoras si aplicamos nuevamente esta representación alternativa a la imagen bLL. Con ello, obtendremos tres imágenes de alta frecuencia que podrán codificarse de forma muy eficiente y una nueva imagen de baja frecuencia (cada vez con menor número de píxeles) que deberemos codificar con precisión. La idea de aplicar este procedimiento a múltiples resoluciones se esquematiza en la figura siguiente.



7.2.2 Implementación compresión de imágenes en .NET C#.

En este apartado vamos a incluir la implementación de la transformada de 1 dimensión wavelet para imágenes tanto en blanco y negro como en color, esta aplicación desarrollada en .NET C# nos permite dada una foto obtener la transformada wavelet de una dimensión obteniendo 4 imágenes a través de las cuales en cualquier momento podemos recuperar la foto madre. Este mecanismo nos permite obtener una representación de la imagen original comprimida.





7.2.3 Aplicaciones de la transformada de wavelet en mejora de imágenes.

En este apartado veremos una forma de enfatizar los contornos de la imagen mediante la descomposición espectral de las imágenes. La idea básica consiste en repetir el proceso de reconstrucción de la imagen pero utilizando las componentes de alta frecuencia enfatizadas. El proceso de reconstrucción es el mismo que el expuesto en apartados anteriores por lo que se omiten los detalles. Las componentes de alta frecuencia se han multiplicado por un factor 2 para que el efecto de enfatización de contornos resulte muy evidente.

```
bLLin=zeros(256,128);bLLin(1:2:256,1:128)=bLL;
```

```

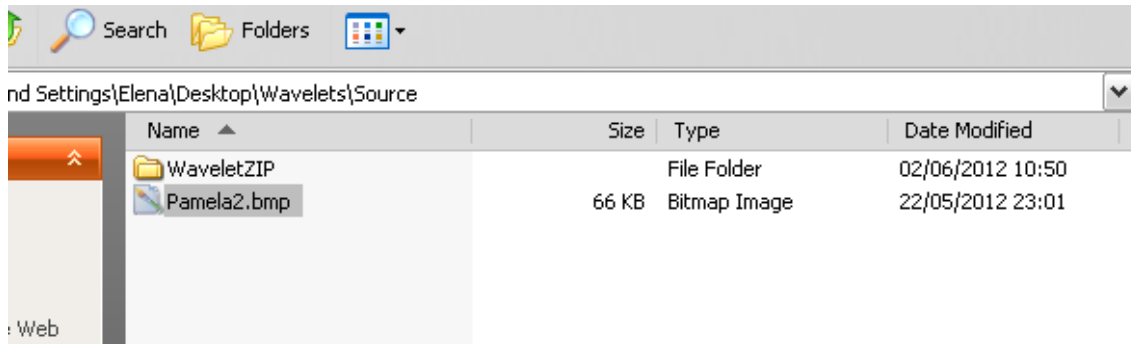
bLHin=zeros(256,128);bLHin(1:2:256,1:128)=2*bLH;
bHLin=zeros(256,128);bHLin(1:2:256,1:128)=2*bHL;
bHHin=zeros(256,128);bHHin(1:2:256,1:128)=2*bHH;
bL1=filter2(h1',bLLin)+filter2(h2',bLHin);
bH1=filter2(h1',bHLin)+filter2(h2',bHHin);
bLin=zeros(256,256); bLin(1:256,1:2:256)=bL1;
bHin=zeros(256,256); bHin(1:256,1:2:256)=bH1;
bra = filter2(h1,bLin)+filter2(h2,bHin);
imshow(bra/max(max(bra)));
figure(2); imshow(b);

```

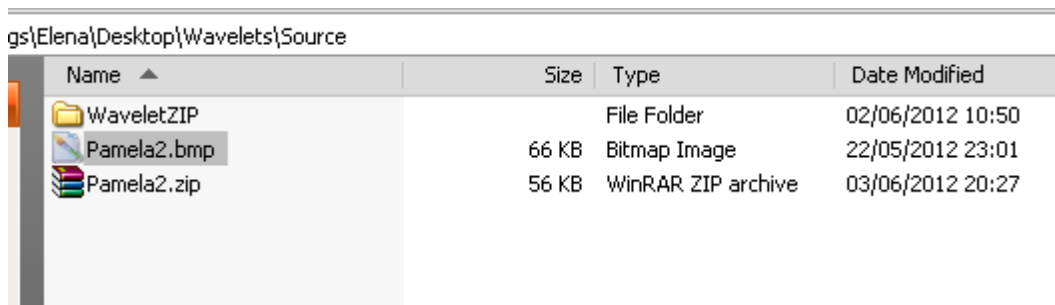


7.2.4 Compresor Wavelet vs ZIP.

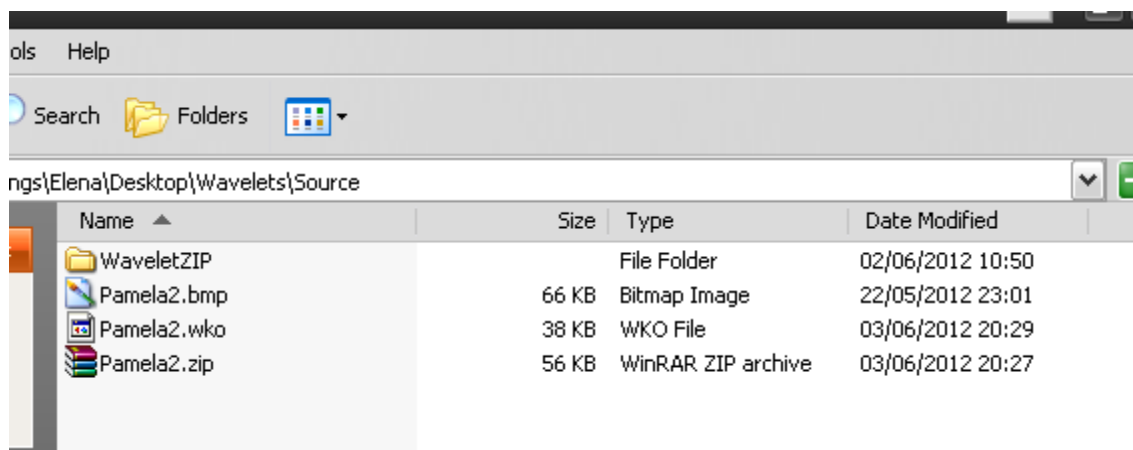
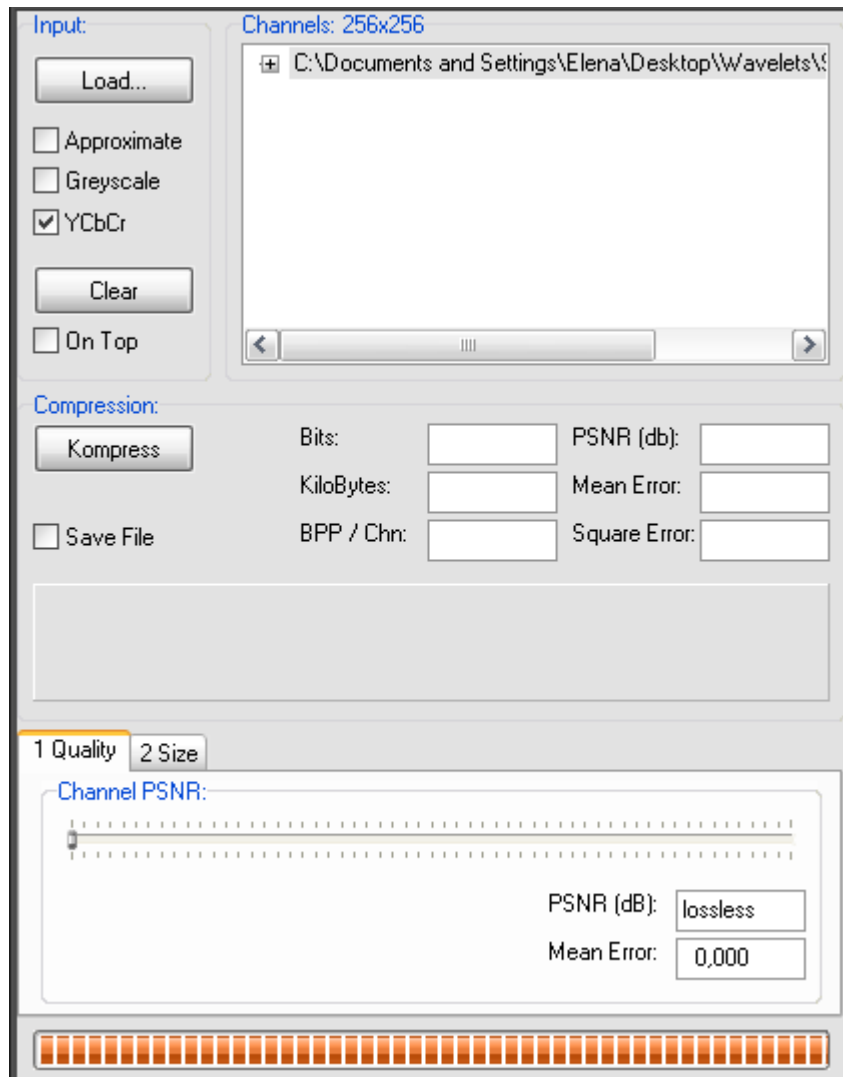
En este apartado vamos a realizar una comparación entre la herramienta ZIP actualmente utilizada en el mercado para la compresión de datos así como la herramienta wavelet_tool_compressor. Vamos a tomar como referencia una la imagen Pamela2.bmp 66KB formato bitmap.



Vamos a comprimir dicha imagen con la herramienta Zip obteniendo un archivo Zip de 56KB

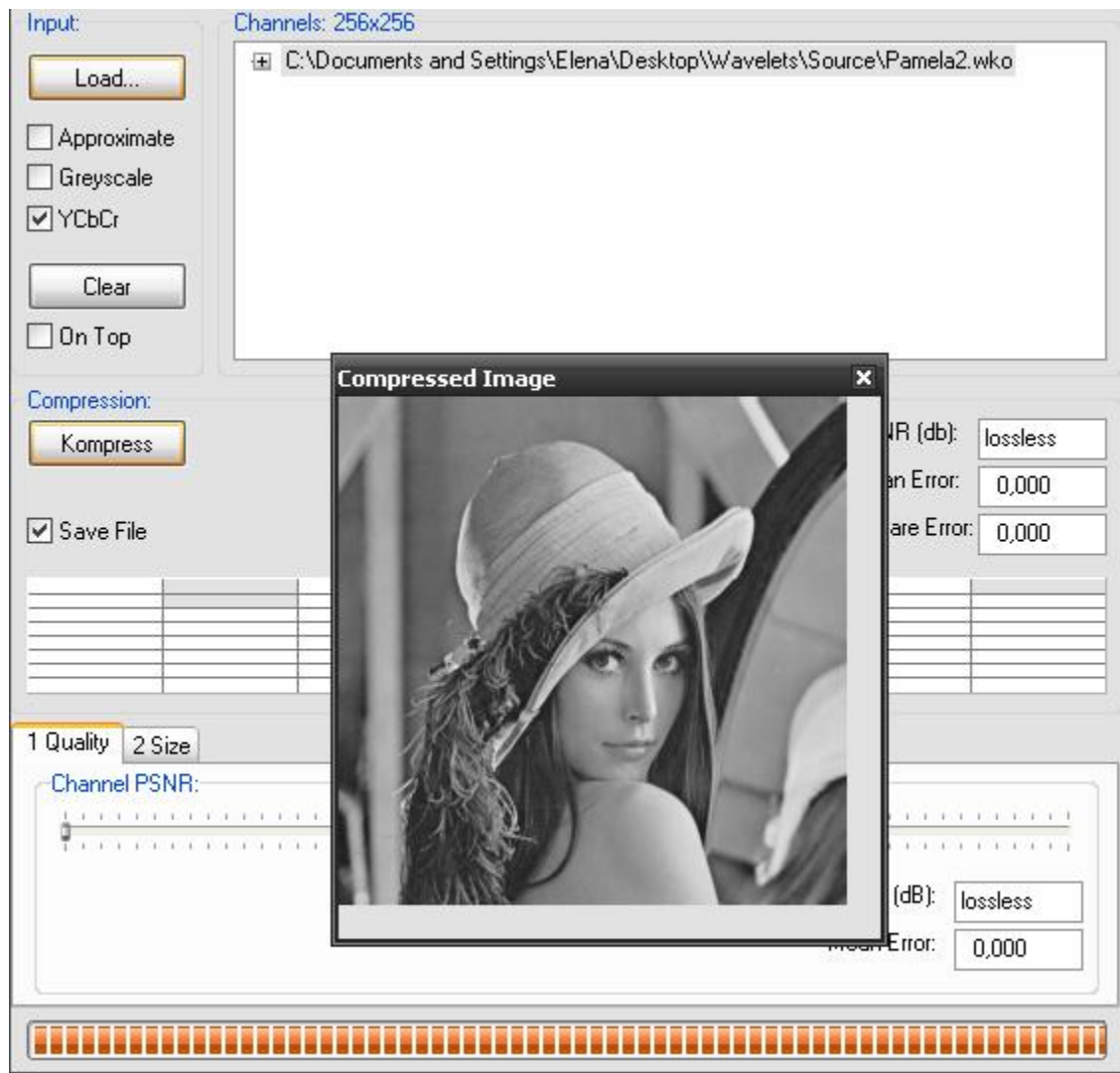


Ahora vamos a comprimir esa misma imagen a través de la herramienta wavelet_tool_compressor obteniendo el archivo Pamela2.wko de 38KB.



Como podemos ver la herramienta que utiliza wavelet nos permite obtener una mejor compresión casi del **50%**. Teniendo en cuenta que en cualquier momento a través del archivo Pamela2.wko podemos recuperar la imagen original.

Reconstrucción de la imagen original a través del fichero Pamela2.wko.



7.2.5 JPEG vs JPEG2000

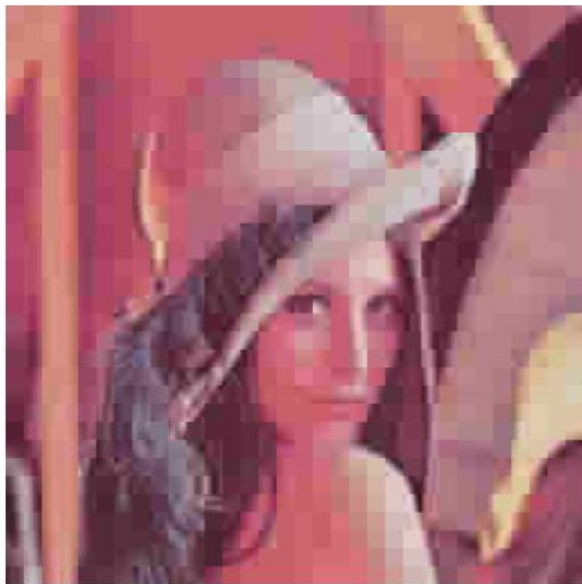
El algoritmo convencional JPEG utiliza Fourier frente al JPEG2000 aprobado en 2003 el cual utiliza Wavelet. En este apartado vamos a realizar una comparación entre ambos algoritmos. A continuación hemos seleccionado dos fotos y hemos comprimido ambas fotos utilizando JPEG y JPEG2000.



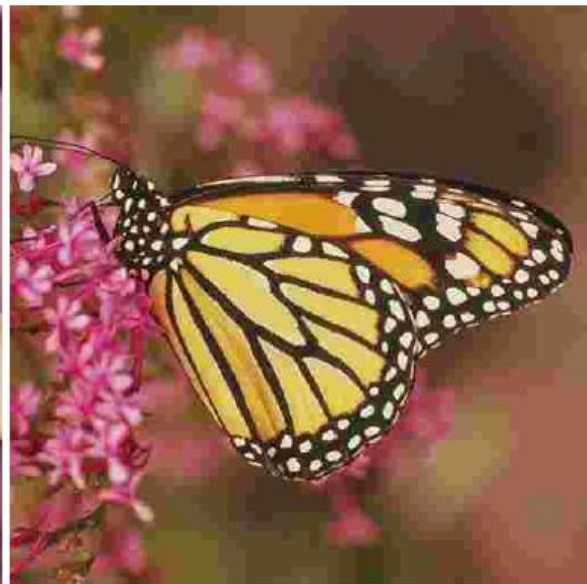
(a) Lena Original, 512x512x24b



(b) Monarch Original, 512x512x24b



(c) Lena JPEG (0.2 bits/pixel)



(d) Monarch JPEG (0.1 bits/pixel)



(e) Lena JPEG2000 (0.2 bits/pixel)



(f) Monarch JPEG2000 (0.1 bits/pixel)

Podemos ver que JPEG divide la imagen original en bloques 8x8 lo que provoca una peor resolución que JPEG2000 que trabaja a nivel de imagen entera. Pudiendo concluir que con JPEG2000 obtenemos una mejor resolución pero el tiempo de procesamiento y la memoria necesaria aumentan.

8 Conclusiones

El análisis de Fourier ha cambiado la forma en que la ciencia y la ingeniería han percibido el mundo, puesto que ha facilitado el florecimiento del tratamiento digital de señales con sus amplias aplicaciones en las comunicaciones, compresión de datos y el análisis de los mismos.

Debido a su limitación en el análisis tiempo frecuencia su detalle de ruido y los eventos que pudieran presentarse, el análisis de Fourier resulta limitadamente apropiado por lo que se presenta la aparición natural de nuevas y más sofisticadas herramientas como son las Wavelets.

9 Bibliografía

A continuación describimos la bibliografía utilizada en el desarrollo del trabajo por orden de relevancia:

[1] Ten Lectures On Wavelets. Ingrid Daubechies. Rutgers University and AT&T Bell Laboratories.

- [2] Wavelets Algorithms & Applications. Yves Meyer Ceremade and Institut Universitaire de France.
- [3] BURKE HUBBARD, Barbara. The world according to wavelets, 261 páginas, Weslesey masachussetts.
- [4] Kenneth R. Casteleman, Digital Image Processing, Prentice Hall, Upper Saddle River, New-Jersey.
- [5 BUR98] Burrus, C.S., Gophinath, R., Guo, H., Introduction to Wavelet and Wavelet Transforms, Prentice Hall, New Jersey 1998.
- [6 CHU97] Chui, C.K., Wavelets: A Mathematical Tool for Signal Processing. SIAM, Filadelfia 1997.
- [7] Murray Spiegel, Teoria y problemas de Análisis de Fourier. McGraw-Hill serie de compendios Schaum, 1981.
- [8] H. Aimar, A. Bernardis and I. Hern´andez, Bases de Onditas, Cuadernos de Matem´atica y Mec´anica, PEMA (Conicet) - CIMEC (INTEC - CONICET - UNL), Santa F´e, 1998.
- [9] Ruel Churchill. “Series de Fourier y Problemas de Contorno”. McGraw-Hill, 1978.
- [10] Tesis doctoral Alfonso Fern´andez Sarria Universidad Polit´ecnica de Valencia. Valencia Febrero 2007.
- [11] Tesis Doctoral Computaci3n Paralela de la Transformada Wavelet. Aplicaciones de la Transformada Wavelet al ´Algebra Lineal Num´erica. Liesner Acevedo Mart´ınez.
- [12] Compression of images, Fourier and wavelets. Radbound University Nijmegen. Maurice Samulski (m.samulski@iphicles.com)
- [13] Scientia et Technica A˜no XIII, No 34, Mayo de 2007. Universidad Tecnol3gica de Pereira. ISSN 0122-1701. DEL AN´ALISIS DE FOURIER A LAS WAVELETS AN´ALISIS DE FOURIER.

10 Referencias Internet

- [1] Diego Santa Cruz, Touradj Ebrahimi, Joel Askelof, Mathias Larsson, and Charilaos Christopoulos, “An analytical studi of JPEG 2000 functionalities”, Swiss Federal Institue of Technology and EricssonResearch ISO/IEC JTC1/SC29/WG1. Available form <http://www.jpeg.org/public/wg1n1816.pdf>.
- [2] Wikipedia (<http://en.wikipedia.org/wiki/Wavelet>)
- [3] Aboufadel, E. y Schlicker, S. The Discovering Wavelets WEB Site. Grand Valley State University [http:// faculty.gvsu.edu/aboufade/web/dw.html](http://faculty.gvsu.edu/aboufade/web/dw.html)

[4] An introduction to Wavelets. Institute of Electrical and Electronics Engineers, Inc.
<http://www.amara.com/IEEEwave/IEEEwavelet.html>

[5] Wavelet Digest (<http://www.wavelet.org/>)

[6] Wavelets made Simple (<http://www.ee.ryerson.ca/~jsantarc/html/theory.html>)

[7] OpenSource Wavelet C Code (<http://www.herbert.the-little-red-hairedgirl.org/en/software/wavelet/>)

[8] Filter Coefficients of Popular Wavelets
 (<http://www.mathworks.com/matlabcentral/files/5502/Filter%20Coefficients%20to%20Popular%20Wavelets.pdf>)

[9] An Introduction to Wavelets (<http://www.amara.com/IEEEwave/IEEEwavelet.html>)

[10] JPEG http://es.wikipedia.org/wiki/Joint_Photographic_Experts_Group

[11] JPEG Org FAQ <http://www.jpeg.org/faq.phtml> & <http://www.jpeg.org/jpeg/>

[12] W3 Org Graphics <http://www.w3.org/Graphics/JPEG/itu-t81.pdf>

11 Apéndices

Ejemplo de descomposición de una imagen.



Wavelet Daubechies

Tabla con los coeficientes filtros de descomposición y reconstrucción correspondientes a la Familia Wavelet Daubechies:

n	Daubechies 1	Daubechies 2	Daubechies 3	Daubechies 4
0	14.14213e-001	4.82962e-001	3.32670e-001	2.30377e-001
1	14.14213e-001	8.36516e-001	8.06891e-001	7.14846e-001
2	-	2.24143e-001	4.59877e-001	6.30880e-001
3	-	-1.29409e-001	-1.35011e-001	-2.79837e-002
4	-	-	-8.54412e-002	-1.87034e-001
5	-	-	3.52262e-002	3.08413e-002
6	-	-	-	3.28830e-002
7	-	-	-	-1.05974e-002
8	-	-	-	-
9	-	-	-	-
10	-	-	-	-
11	-	-	-	-
12	-	-	-	-
13	-	-	-	-
14	-	-	-	-
15	-	-	-	-

n	Daubechies 5	Daubechies 6	Daubechies 7	Daubechies 8
0	1.60102e-001	1.11540e-001	7.78520e-002	5.44158e-002
1	6.03829e-001	4.94623e-001	3.96539e-001	3.12871e-001
2	7.24308e-001	7.51133e-001	7.29132e-001	6.75630e-001
3	1.38428e-001	3.15250e-001	4.69782e-001	5.85354e-001
4	-2.42294e-001	-2.26264e-001	-1.43906e-001	-1.58291e-002
5	-3.22448e-002	-1.29766e-001	-2.24036e-001	-2.84015e-001
6	7.75714e-002	9.75016e-002	7.13092e-002	4.72484e-004
7	-6.24149e-003	2.75228e-002	8.06126e-002	1.28747e-001
8	-1.25807e-002	-3.15820e-002	-3.80299e-002	-1.73693e-002
9	3.33572e-003	5.53842e-004	-1.65745e-002	-4.40882e-002
10	-	4.77725e-003	1.25509e-002	1.39810e-002
11	-	-1.07730e-003	4.29577e-004	8.74609e-003
12	-	-	-1.80164e-003	-4.87035e-003
13	-	-	3.53713e-004	-3.91740e-004
14	-	-	-	6.75449e-004
15	-	-	-	-1.17476e-004

Comandos de MatLab utilizados para el procesamiento de imágenes.

Cargar imágenes en matrices:

```
A=double(imread(NombreImagen,Formato));
```

Visualizar imágenes a partir de una matriz:

```
imshow(uint8(A))
```

Visualizar más de una imagen en una misma ventana:

```
subplot(2,2,1);imshow(uint8(A))
```

```
subplot(2,2,2);imshow(uint8(B))
```

```
subplot(2,2,3);imshow(uint8(C))
```

```
subplot(2,2,4);imshow(uint8(D))
```

Transformada discreta de Fourier en dos dimensiones

```
Fa=fft2(A);
```

Antittransformada discreta de Fourier en dos dimensiones

```
A=ifft2(Fa);
```

Visualización de la transformada discreta de Fourier

```
mesh(abs(Fa))
```

Filtro Digital: $Y = \text{filter2}(B, X);$

Madrid Junio 2012

Autorizo a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, los contenidos audiovisuales incluso si incluyen imágenes de los autores, la documentación y/o el prototipo desarrollado.

Fdo.
Elena Rupérez Cerezo

